Matlab与化工数值计算

第一讲 简介与基本数学运算

隋志军

化工学院软件应用教科组

联系方式：zhjsui@ecust.edu.cn

64252169

实验16楼605室

化学工程师的任务

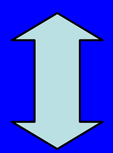


工厂运行

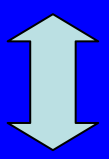
系统特性



数学模型



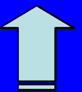
过程设计

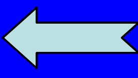


数学模型



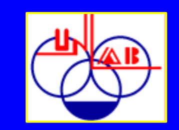
反应特征
过程特性
设备特性





工艺开发

化学工程专业数学模型类型



非线性方程（组）

$$\frac{1}{\sqrt{f}} = \frac{4.0}{n'^{0.75}} \lg(\mathrm{Re}_{gen} \; f^{1-n'/2}) - \frac{0.4}{n'^{1.2}}$$

常微分方程（组）

$$f''' + ff'' + \beta(1 - (f')^2) = 0$$

偏微分方程（组）

$$\frac{\partial u_1}{\partial t} = 0.024 \frac{\partial^2 u_1}{\partial x^2} - F(u_1 - u_2)$$

$$\frac{\partial u_2}{\partial t} = 0.170 \frac{\partial^2 u_2}{\partial x^2} + F(u_1 - u_2)$$

非线性模型，难以获得解析解，必须采用数值解法

模型的数值解法是应用数学的一个分支，通常称为计算数学（数值分析，数值方法）

化学工程常用软件



数学软件:

- Matlab
- Mathematica
- Mathcad
- Maple
- Staticstica

化工模拟软件:

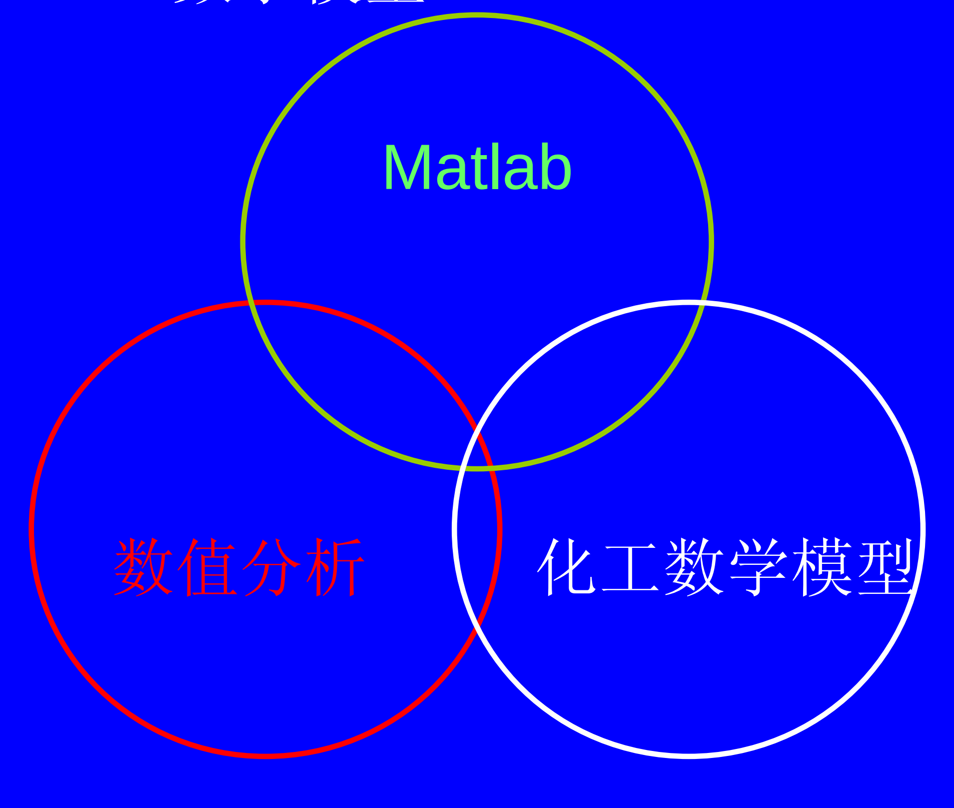
- ➢PRO/II (SimSci)
- ➢AspenPlus
- ➢ChemCAD
- ➢Flowtran
- ➢Superpro Designer
- ➢Fluent
- ➢CHEMKIN

本课程的学习目的



学会Matlab的使用，可以利用Matlab求解较为复杂的化工数学模型

对于数值分析的内容不过多涉及，只注意数值计算结果的准确性



化工专业知识作为背景，不涉及模型的推导，注重模型求解过程的方法与技巧

本课程基本内容



- 第一讲 Matlab简介与基本数学运算
- 第二讲 非线性方程组求解与迭代法
- 第三讲 矩阵操作与线性方程组求解
- 第四讲 插值、拟合与数值微分、积分
- 第五讲 常微分方程数值解
- 第六讲 偏微分方程数值解
- 第七讲 统计初步与最优化方法

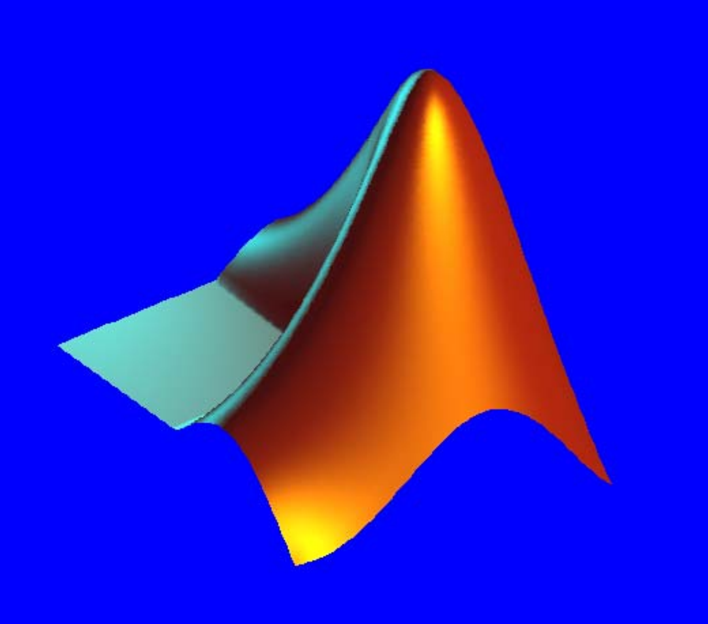
学习本课程的注意事项



- 学好本课程的唯一途径是多上机实践
- 数值计算效率和效果的保证有很多技巧,可以参考数值方法（数值分析）方面的教科书
  - 刘则毅，科学计算技术与Matlab，科学出版社
  - 同济大学计算数学教研室，现代数值数学和计算，同济大学出版社
  - 黄华江，实用化工计算机模拟，化学工业出版社
  - 张志涌，精通Matlab6.5版，北京航空航天大学出版社
- 对于数值计算的结果，应注意分析结果的意义

Matlab简介

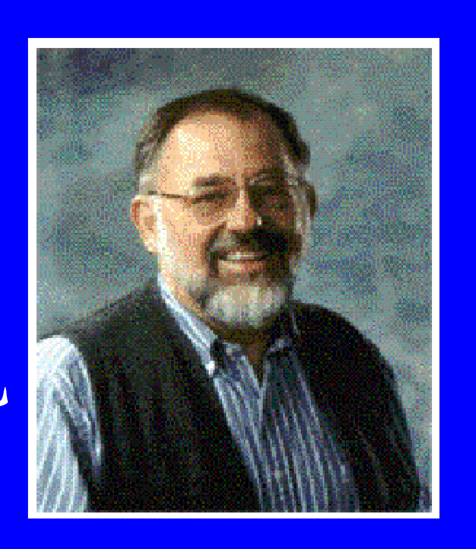




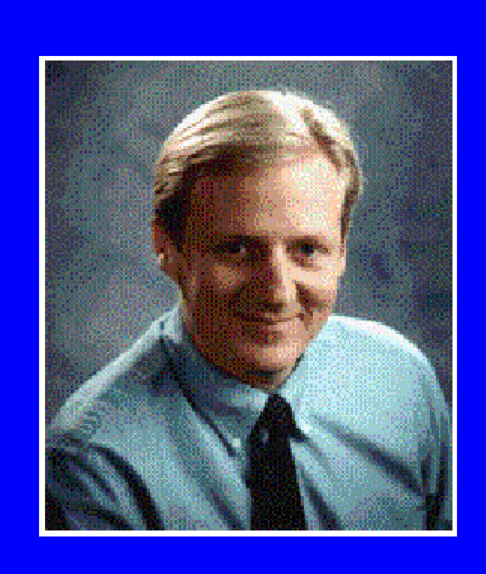
Matlab是Matrix Labotary的缩写，最初是美国新墨西哥大学Moler教授编写的LINPACK和EISPACK接口程序

1984年，MathWorks公司创建，MATLAB正式推向市场

20世纪90年代以来，MATLAB已成为数值计算软件的佼佼者



Prof. Cleve Moler

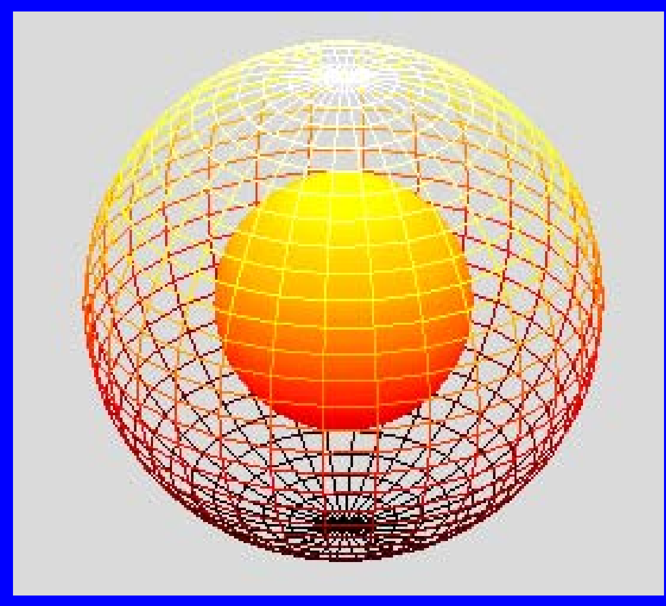


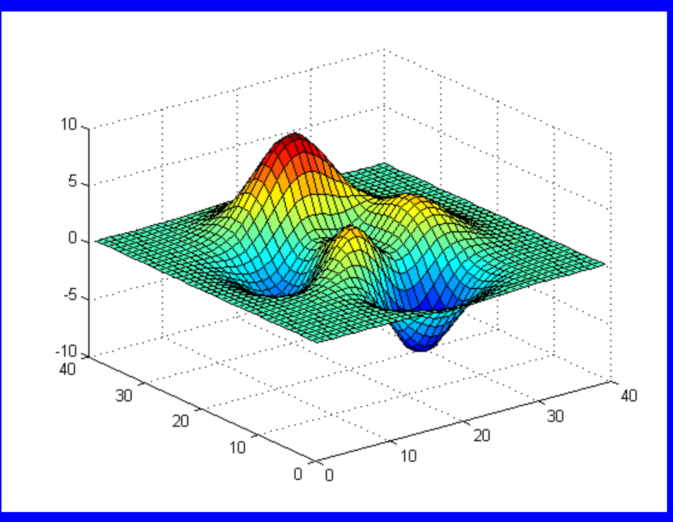
Jack Little

Matlab简介



- MATLAB具有用法简单、灵活、结构性强、延展性好等优点，逐渐成为科技计算、视图交互系统和程序中的首选语言工具。
  - 功能强大的数值运算功能
  - 强大的图形处理能力
  - 高级但简单的程序环境
  - 丰富的工具箱与模块集
  - 易于扩充





开始的问题



计算在1/2英寸不锈钢管中，以2000lb/hr流量输送水，当水的温度为10、20、30、40、50、60、70、80℃时，压降分别为多少？

牛顿流体在不锈钢管中的流动压降可由下式估算：$\Delta P = \dfrac{M^{1.8}\mu^{0.2}}{20000 D^{4.8}\rho}$

其中，摩擦压降，psi/(100英尺等量管长)；M，质量流量，lb/hr；μ，粘度，cP；ρ，密度，lb/ft3，D，管径，inch。

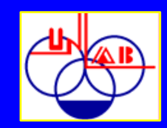
流体密度可由下式描述：$\rho = A \cdot B^{-(1-T/T_C)^n}$

ρ，g/ml；对于水，A = 0.34710；B = 0.2740；Tc = 647.13K；n = 0.28571。

流体粘度由下式描述：$\log_{10}^{\mu} = A + B/T + CT + DT^2$

μ，cP；对于水，A=-10.2158；B=1.7925E3；C = 1.7730E-2；D=-1.2631E-05。

Matlab窗口介绍



MATLAB

File Edit Debug Desktop Window Help

C:\Program Files\MATLAB71\work

Shortcuts How to Add What's New MGFforLOTPD

Workspace

Base

Name	Value	Class

Workspace Current Directory

Command History

Command Window

To get started, select MATLAB Help or Demos from the Help menu.

>>

Start

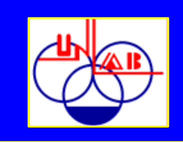
变量空间

当前路径

命令历史

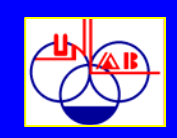
命令窗口

Matlab的通用命令



命令	命令说明	命令	命令说明
cd	显示或改变工作目录	dir	显示目录文件
type	显示文件内容	clear	清除内存变量
clf	清除图形窗口	pack	收集内存碎片，扩大内存空间
clc	清除命令窗口内容	echo	命令窗口信息显示开关
hold	图形保持开关	disp	显示变量或文字内容
path	显示搜索目录	save	保存内存变量到指定文件
load	加载指定文件变量	diary	日志文件命令
quit	退出Matlab	!	调用DOS命令
whos	变量查看		

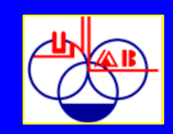
通过Help学习Matlab



在命令窗口中键入>> help，则显示以下内容：

matlab\general - General purpose commands.
matlab\ops - Operators and special characters.
matlab\lang - Programming language constructs.
matlab\elmat - Elementary matrices and matrix manipulation.
matlab\elfun - Elementary math functions.
matlab\specfun - Specialized math functions.
matlab\matfun - Matrix functions - numerical linear algebra.
matlab\datafun - Data analysis and Fourier transforms.
matlab\polyfun - Interpolation and polynomials.
matlab\funfun - Function functions and ODE solvers.
matlab\sparfun - Sparse matrices.
matlab\scribe - Annotation and Plot Editing.
matlab\graph2d - Two dimensional graphs.
matlab\graph3d - Three dimensional graphs.
………

Help + 主题名称



```
>> help ops
 Operators and special characters.

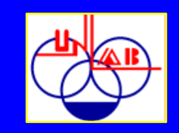
 Arithmetic operators.
   plus        - Plus                                  +
   uplus       - Unary plus                            +
   minus       - Minus                                 -
   uminus      - Unary minus                           -
   mtimes      - Matrix multiply                       *
   times       - Array multiply                        .*
   mpower      - Matrix power                          ^
   power       - Array power                           .^
   mldivide    - Backslash or left matrix divide       \
   mrdivide    - Slash or right matrix divide          /
   ldivide     - Left array divide                     .\
   rdivide     - Right array divide                    ./
```

基本算术运算符



运 算	符 号	运 算	符 号
加	+	减	-
矩阵乘	*	数组相乘	.*
矩阵左除	\	数组左除	.\
矩阵右除	/	数组右除	./
幂次方	^	数组幂次方	.^

Help + 函数名



```
>> help power
 .^  Array power.
    Z = X.^Y denotes element-by-element powers.  X and Y
    must have the same dimensions unless one is a scalar.
    A scalar can operate into anything.

    C = POWER(A,B) is called for the syntax 'A .^ B' when A or B is an
    object.
```

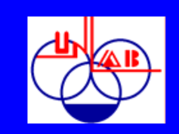
Help+函数名可获得详细的函数使用方法

Matlab语言的标点



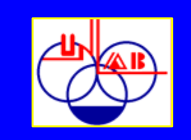
标点	定义	标点	定义
:	向量和矩阵的多种功能	.	小数点及结构体域的访问
;	区分行及取消行显示	…	续行符
,	区分列及函数参数分隔符	%	注释符，百分号
( )	指定运算过程的次序等	!	调用dos操作命令
[]	矩阵定义	=	赋值标记
{}	构成单元数组	‘	字符串标示符

数值类型



- 分类方法一
  - 双精度型　（系统默认类型）
  - 单精度型
  - 带符号整数
  - 无符号整数
- 分类方法二
  - 标量
  - 向量
  - 数组
- 分类方法三
  - 实数
  - 复数

数值的表示



以下表达方式均合法:

345 -99 0.01

1.3e-3 4.5e33

[1 2 3] [1;2;3]

[1 2; 2 11]

3+3i 6-8j

基本数学运算符的使用



计算以下表达式的值：

1) [1 2 3]*[3 2 1]

2) [1 2 3].*[3 2 1]

3) [1 2 3]^2

4) [1 2 3].^2

5) 1+3*2^2

6) (3*2)^2

7) (3*2)^2;

8) (-8)^(1/3)

Matlab的计算器功能

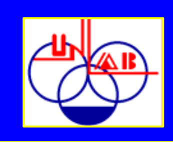


```
>> 2000^1.8*(10^(-10.2158+1.7925e3/283+1.773e-2*283-1.2631e- 5*283^2))^0.2/(20000*0.5^4.8*(0.3471*0.274^(-(1-283/647.13)^0.28574))/0.2323)
```

回车可以得到结果

```
ans =
  287.8245
```

命令的窗口的快捷键



快捷键	作用	快捷键	作用
↑，Crtl+P	回调上一行	Crtl+→	右移一单词
↓，Crtl+N	回调下一行	Crtl+A，Home	移至行首
←，Crtl+B	回移上一字符	Crtl+E，End	移至行末
→，Crtl+F	前移下一字符	Crtl+U，Esc	删除一行
Crtl+←	左移一单词	Crtl+K	从光标删除至行末
Ctrl+C	终止正在运行的程序		

数学函数（elfun）



类型	函　　数	含　　义
三角函数	**sin(x)**	正弦值
	asin(x)	反正弦值
	cos(x)	余弦值
	acos(x)	反余弦值
	tan(x)	正切
指数函数	**exp(x)**	指数运算
	log(x)	自然对数
	sqrt(x)	求平方根
复数函数	**abs(x)**	求绝对值
	imag(x)	取出复数的虚部
	real(x)	取出复数的实部
	conj(x)	复数共轭
数论函数	**round(x)**	四舍五入
	mod(x,y)	求余数
	lcm(x,y)	整数x和y的最小公倍数
	gcd(x,y)	整数x和y的最大公约数

基本数学函数的使用



计算以下表达式的值：

1) sin(30)
2) sind(30)
3) exp([1 2 3])
4) log10([10 100 1000])
5) abs(3+4i)
6) abs(-5)

format命令



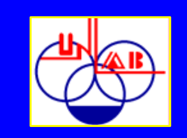
MATLAB命令	含　义	范　例
format short	短格式	3.1416
format short e	短格式科学格式	3.1416e+000
format long	长格式	3.14159265358979
format long e	长格式科学格式	3.141592653589793e+000
format rat	有理格式	355/113
format hex	十六进制格式	400921fb54442d18
format bank	银行格式	3.14

程序的组成



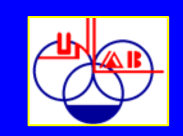
数据输入	变量 →	运算	变量 →	数据输出
键盘输入		数学运算		屏幕输出
文件输入		关系运算		文件输出
		逻辑运算		图形输出
		流程控制		

变量



- 变量的命名方式：
  - 变量名由字母、数字和下划线组成；
  - 变量名中的英文字母大小写是有区别的；
  - 变量名的最大长度是有规定的
    - 不同版本的系统规定不同：19个字符、31或63个字符等
    - 可调用namelengthmax函数得到系统规定长度

变量的使用



```
>>clear                    %删除工作区中所有定义过的变量
>>whos                     %查看当前工作区内变量信息，无显示表示没
    有定义的变量
>> xy=1; yx=2;             %对变量赋值
>> xy                      %查看变量xy的当前数值
xy =
    1
>> whos
  Name      Size                Bytes  Class
  xy        1x1                 8  double array
  yx        1x1                 8  double array
Grand total is 2 elements using 16 bytes
>> clear xy yx             %删除变量xy及yx
>> whos
>> xy                      %这时变量xy已经不存在了
??? Undefined function or variable 'xy'.
```

MATLAB系统的特殊变量和常数



特殊变量	意　　　义
ans	如果用户未定义变量名，系统用于计算结果存储的默认变量名
pi	圆周率π（**= 3.1415926...**）
inf或Inf	无穷大∞值
eps	浮点运算的相对精度**2^(-52)**
realmax	最大的正浮点数，**2^(1024)-1**
realmin	最小的正浮点数，**2^(-1022)**
NaN或nan	不定量
i或j	虚数单位
nargin	函数输入参数个数
nargout	函数输出参数个数
lasterr	存放最新的错误信息
lastwarn	存放最新的警告信息

MATLAB数据类型



- 数值（标量，向量，数组）
- 字符串
- 单元数组（cell array）
- 结构体（structure）
- 函数句柄

向量的生成



1) 直接输入向量

格式上要求向量元素需要用“[]”括起来，元素之间可以用空格、逗号或分号分隔。用空格和逗号分隔生成行向量，用分号生成列向量。

2) 利用冒号生成向量

冒号表达式的基本形式为：x = x0:step:xn

若step = 1，则此项输入可以忽略。

3) linspace函数

可以使用linspace函数生成线性等分向量:

y = linspace(x1,x2)　　生成（1*100）维行向量，y(1)=x1，y(100)=x2

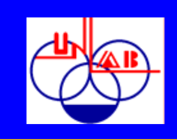
y = linspace(x1,x2, n)　生成（1*n）维行向量，y(1)=x1，y(n)=x2

4) logspace函数

logspace用于生成对数等分向量，格式如下:

y = logspace(x1,x2,n) 生成（1*n）维对数等分向量，y(1)=10^x1，y(n)=10^x2；n可以省略，此时其默认值为50。

向量的运算



1) 向量加减与数加减

向量的加减与数加减的形式与普通标量加减相同

2) 向量的点积、叉积与混合积的实现

点积：向量的点积由函数dot实现。dot(a,b)返回向量a和b的数量点积，其中a，b必须同维。

叉积：叉积由cross函数实现。向量a,b必须为三维向量

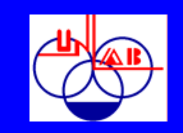
混合积：可由以下命令实现，dot(a,cross(b,c))

3)向量的数乘、数组乘和向量乘

例：当a = [1:1:3]; b=[2:2:6]时，以下命令的运行结果是什么？

1)>>a1=2*a　　　2) >> a2=a.*b　　　3) >>a3=a*b

字符串类型



- 字符串：包含在一对单引号中的字符集合

```
>> s='hello, MATLAB'              %定义字符串变量s
s =
hello, MATLAB

>> whos
  Name      Size                   Bytes  Class
  s         1x13                      26  char array
Grand total is 13 elements using 26 bytes
```

单元数组（Cell Array）



单元数组是MATLAB数组的一种特殊数据类型，它用于保存不同类型和/或不同大小的数据。

三种直接赋值方式

1. 单元下标用括号“（）”括起来，而单元的内容用“{}”括起来，如:

```
>>clear all
>>a(1,1)={[1 2;3 4]};
>>a(1,2)={[0 1]};
>>a(2,1)={'Hello'};
>>a(2,2)={2+3i}
```

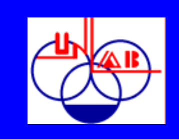
2.单元下标用“{}”括起来，而赋值语句等式右边的单元内容用“[]”括起来:

```
>>a{1,1}=[1 2;3 4];
>>a{1,2}=[0 1];
>>a{2,1}='Hello';   %右边只有一个元素时可省略去“[]”
>>a{2,2}=2+3i
```

3. 直接使用{}

```
>>a={[1 2;3 4],[0 1],'Hello',2+3i}
```

单元数组的操作



显示单元数组的命令

>>a　　　　　% 显示单元数组a的信息

>>celldisp(a)　　% 显示单元数组a的完整内容

先使用函数cell()创建空的单元数组，然后再赋值:

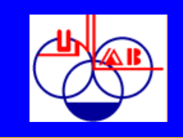
>>b=cell(2,3)

赋值方法同直接赋值方式。

对单元数组元素的操作

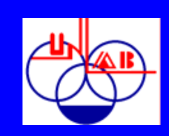
>>c=a{1,2}　　% 将单元数组a的{1,2}元素赋给变量c，注意是“{}”,而不是“()”。

结构体



- 与C语言类似，MATLAB结构体用于存取相关的数据，它由一组称为域（fields）的成员变量（向量）构成，每一个域可以为不同的MATLAB数据类型。
- 结构数组的定义有两种方法，一种是直接赋值，另一种是使用strct()函数。

结构体的赋值



```
>>student.name='Zhang Jun';
>>student.major='Chemical Engineering';
>>student.subject=['英语 ','政治 ','数学 ','化工原理 ','物理化学 '];
>>student.entrance_exam=[62 68 72 82 90];

>>student(2).name='Li Xia';
>>student(2).major='Chemical Engineering';
>>student(2).subject=['英语 ','政治 ','数学 ','化工原理 ','物理化学 '];
>>student(2).entrance_exam=[60 72 68 85 88];
```

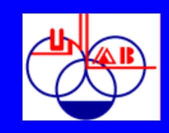
```
struct_array_name=structure('field1',values1,'field2',values2,…)
Student=struct('name','Zhang Jun', 'major','Chemical Engineering')
```

管道压降的计算



```
T=[283:10:353];
M=2000;D=0.5;
density.A=0.3471;density.B=0.274;density.Tc=647.13;density.n=0.28571;
Rho=(density.A.*density.B.^(-(1-T./density.Tc).^density.n))/0.2323;
mu.A=-10.2158;mu.B=1.7925e3;mu.C=1.773e-2;mu.D=-1.2631e-5;
mu=10.^(mu.A+mu.B./T+mu.C.*T+mu.D.*T.^2);
deltP=(M^1.8)*(mu.^0.2)./(20000*D^4.8.*Rho)
```

函数文件和Script文件



Script文件

Script仅仅是一连串可执行的MATLAB命令，它具有全局性

Script文件中不能定义函数

函数

函数定义的一般格式:

```
function [y1,y2,…,yn] = FuncName(x1,x2,…,xn)   %函数声明语句
                    y1 = …      %（表达式1）
                    y2 = …      %（表达式2）
                                 …
                    yn = …      %（表达式n）
```

其中，输入参数为x1,x2,…,xn,输出参数为y1,y2,…,yn。各参数可以是标量、向量或矩阵。

脚本编辑窗口



Editor - Untitled

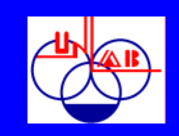
File Edit Text Cell Tools Debug Desktop Window Help

Stack: Base

1

script | Ln 1 | Col 1 | OVR

函数文件的编写



编写一个函数，计算本章开始问题中流体的粘度，函数要求输出粘度的计算值：

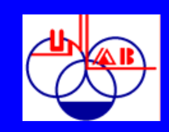
1. 函数声明语句： function vis=viscosity()
2. 变量的传递
   1. 通过调用函数传递
      - vis=viscosity(A,B,C,D,T)
   2. 通过全局变量传递
      - 利用global命令，在主函数和子函数中予以声明
3. 编写表达式

函数的调用



1. 在调用函数的主函数中，直接采用函数名调用
2. 通过函数句柄调用

管道压降的计算函数



```
function Cha1demo4_7
global mu T
T=[283:10:353];
M=2000;D=0.5;
density.A=0.3471;
density.B=0.2740;
density.Tc=647.13;
density.n=0.28571;
Rho=(density.A.*density.B.^(-(1-T./density.Tc).^density.n))/0.2323
mu.A=-10.2158;mu.B=1.7925e3;mu.C=1.773e-2;mu.D=-1.2631e-5;
Mu=viscosity
deltP=(M^1.8)*(Mu.^0.2)./(20000*D^4.8.*Rho)
%---------------------------------------------------------------
function vis=viscosity
global mu T
vis=10.^(mu.A+mu.B./T+mu.C.*T+mu.D.*T.^2);
```

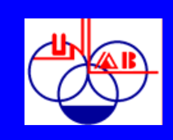
函数句柄



- 创建一个函数句柄，可用于保存函数的所有信息，以便将来对它进行调用。
- 函数句柄可以作为参数传递给其他函数，或与feval函数一起使用，以调用该函数句柄所属的函数。
- 使用函数句柄还可以减少定义函数的文件个数，改善重复操作的性能，保证函数计算的可靠性。
- funhandle = @function_name

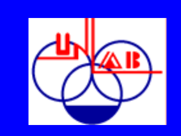
  %function_name为用户指定的函数名

管道压降的计算函数



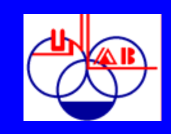
```
function Cha1demo4_5
T=[283:10:353];
M=2000;D=0.5;
density.A=0.3471;
density.B=0.274
density.Tc=647.13;
density.n=0.28571;
Rho=(density.A.*density.B.^(-(1-T./density.Tc).^density.n))/0.2323
mu.A=-10.2158;mu.B=1.7925e3;mu.C=1.773e-2;mu.D=-1.2631e-5;
Mu=feval(@viscosity,mu,T)%Mu=viscosity(mu,T)
deltP=(M^1.8)*(Mu.^0.2)./(20000*D^4.8.*Rho)
%-----------------------------------------------------------------
function vis=viscosity(mu,T)
vis=10.^(mu.A+mu.B./T+mu.C.*T+mu.D.*T.^2);
```

内联函数（inline function）



- 内联函数是Matlab提供的一个对象，它的表现和函数文件一样，但内联函数的创建比较容易
- 内联函数的创建
  - ✓inline('CE')
  - ✓inline('CE',arg1,arg2,...)
  - ✓inline('CE',n)
- 涉及内联函数性质的指令
  - ✓class(inline_fun)　　内联函数类型
  - ✓char(inline_fun)　　给出内联函数计算公式
  - ✓argnames(inline_fun) 给出内联函数的输入变量
  - ✓vectorize(inline_fun)　使内联函数适用于数组运算
- Matlab中许多“泛函”函数都是采用inline，从而具备了适应各种被处理函数形式的能力

内联函数的应用

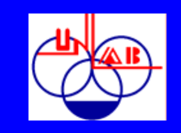


```
F1=inline('sin(rho)/rho')
f1=F1(2)

FF1=vectorize(F1)
xx=[0.5,1,1.5,2];ff1=FF1(xx)

G2=inline('a*exp(x(1))*cos(x(2))','a','x')
g1=G2(2,[-1,pi/3])
```

匿名函数（anonymous function）



- 匿名函数用于在命令行、函数文件或script文件中创建简单形式的函数，避免另外定义新的函数
- 匿名函数的定义形式

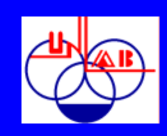
  f=@(arglist)expression

```
f=@(x) x.^2
a=f(5)
```
结果： a = 25

```
f=@(x) x.^2;
g=@(x) 3*x;
h=@(x) g(f(x));
h(3)
```
结果： ans = 27

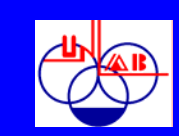
```
alpha=0.9;
f=@(x) sin(alpha*x);
f(pi)
```
结果： ans = 0.3090

数据输入和输出



- 数据输入
  - –利用M文件产生数据文件
  - –用Load命令从MAT文件或文本文件读取数据
  - –用fscanf函数
  - –用提示输入函数input
  - –dlmread, importdata, xlsread函数
- 数据输出
  - –用Save命令
  - –用fprintf函数
  - –用函数disp()将结果输出至屏幕
  - –dlmwirte,xlswrite函数
  - – 图形输出

Matlab二维图形



1	数据准备： ●选定所要表现的范围 ●产生自变量采样向量 ●计算相应的函数值向量	t=pi*(0:100)/100; y=sin(t).*sin(9*t);
2	选定图形窗及子图位置 ●缺省时，打开Figure No.1，或当前窗，当前子图 ●可用指令指定图形窗号和子图号	figure(1)　%指定1号图形窗 subplot(2,2,3)　%指定一个具有2行2列子图图形窗中的3号子图
3	调用（高层）绘图指令；在指令中设置线型、色彩、数据点型	plot(t,y,'b-')　%用蓝色实线画曲线
4	设置轴的范围与刻度、坐标分格线	axis([0,pi,-1,1])　%设置轴的范围 grid on　%画坐标分格线
5	图形注释： 图名、坐标名、图例、文字说明	title('调制波形')　%图名 xlabel('t');ylabel('y')　%轴名 legend('sin(t)','sin(t)sin(9t)')　%图例 text(2,0.5,'y=sin(t)sin(9t)')　%文字说明
6	图形的精细修饰（图柄操作）： ●利用对象属性值进行设置 ●利用图形窗工具条进行	set(h,'MarkerSize',10)　%设置数据点大小

函数Plot基本调用格式



1) plot(X,'s')

- X为实向量时，以该向量元素的下标为横坐标、元素值为纵坐标画一条连续曲线
- X是实矩阵时，则按列绘制每列元素值相对其下标的曲线，曲线数目等于X的列数
- X是复数矩阵时，则按列分别以元素的实部和虚部为横、纵坐标绘制多条曲线
- 's'是用来控制线型、色彩、数据点型的选项字符串。s可以缺省，此时曲线按Matlab默认设置绘制。s的取值见下节

2) plot(X,Y,'s')

- X、Y是同维向量时，绘制以X、Y为横、纵坐标的曲线
- X是向量，Y是有一维与X同维的矩阵时，则绘出多根不同色彩的曲线。曲线数等于Y的另一维，X作为这些曲线共同的横坐标
- X是矩阵，Y是向量时，情况与上相同，只是曲线都以Y为共同纵坐标
- X、Y是同维矩阵时，则以X、Y对应列元素为横、纵坐标分别绘制曲线，曲线条数等于矩阵的列数。
- 's'的意义，与上相同。

3) plot(X1,Y1,'s1',X2,Y2,'s2',…)

- 此格式中，每个绘线“三元组”（X,Y,'s'）的结构和作用，与上相同。不同“三元组”之间没有约束关系。

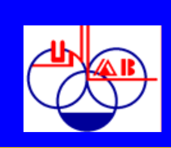
曲线的色彩、线型和数据点型貌



线型	符号	-		:		-.		--	
	含义	实线		虚线		点划线		双划线	
色彩	符号	b	g	r	c	m	y	k	w
	含义	蓝色	绿色	红色	青色	品红	黄色	黑色	白色

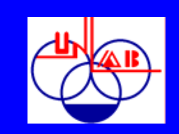
符号	含义	符号	含义
.	实心黑点	d	菱形
+	十字	h	六角星符
*	八线符	o	空心圆圈
^	上三角	p	五角星符
>	右三角	s	方块
<	左三角	x	叉字符
v	下三角		

坐标、刻度和分格线控制



指令	含义	指令	含义
axis auto	缺省设置	axis equal	纵横轴采用等长刻度
axis manual	当前坐标范围不变	axis fill	manual方式下起作用，使坐标充满整个绘图区
axis off	取消轴背景	axis image	纵、横轴采用等长刻度，坐标框紧贴数据范围
axis on	使用轴背景	axis normal	缺省矩形坐标系
axis ij	矩阵式坐标，原点在左上方	axis square	产生正方形坐标系
axis xy	普通直角坐标	axis tight	把数据范围设为坐标范围
axis(V) V=[x1,x2,y1,y2] V=[x1,x2,y1,y2,z1,z2]	人工设定坐标范围。设定值：二维，4个；三维，6个	axis vis3d	保持高宽比不变

图形标识



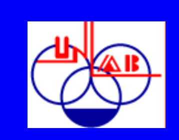
图形标识命令	调用格式
图名（title）	TITLE(‘text’) TITLE('text','Property1',PropertyValue1,'Property2',PropertyValue2,...)
坐标轴（label）	XLABEL('text’) XLABEL('text','Property1',PropertyValue1,'Property2',PropertyValue2,...)
图形注释（text）	TEXT(X,Y,'string')
图例（legend）	LEGEND(string1,string2,string3, ...)

例题



```
T=[283:10:353];
M=2000;D=0.5;
density.A=0.3471;density.B=0.274;density.Tc=647.13;density.n=0.28571;
Rho=(density.A.*density.B.^(-(1-T./density.Tc).^density.n))/0.2323;
mu.A=-10.2158;mu.B=1.7925e3;mu.C=1.773e-2;mu.D=-1.2631e-5;
mu=10.^(mu.A+mu.B./T+mu.C.*T+mu.D.*T.^2);
deltP=(M^1.8)*(mu.^0.2)./(20000*D^4.8.*Rho)plot(T,deltP,'b-o')
title('The pressure drop vs Temperature')
xlabel('Temperature(^oC)')
ylabel('Pressure drop (psi/equivalent feet of pipe)')
```

图形标识的精细控制



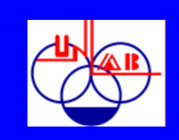
字体	指令	arg 取值	举例	
			示例举例	效果
名称	\fontname{arg}	所有 windows 字库字体	'\fontname{courier}Example1' '\fontname{隶书}范例 2'	Example1 范例2
风格	\arg	bf(黑体), it(斜体), rm（正体）	'\btExample3' '\itExample4'	**Example3** *Example4*
大小	\fontsize{arg}	正整数，缺省值为 10 磅	'\fontsize{14}Example5' '\fontsize{6}Example6'	Example5 Example6
上标	^{arg}	任何合法字符	'\ite^{-t}sint'	$e^{-t}sint$
下标	_{arg}	任何合法字符	'x~{\chi}_{\alpha}^{2}(3)'	$x\sim\chi_{\alpha}^{2}(3)$

句柄图形的结构



- Root
  - Figure
    - Uicontrol
    - Axes
      - Line
      - Surface
      - Rectangle
      - Patch
      - Image
      - Text
      - Light
    - Uimenu

句柄图形的访问与操作

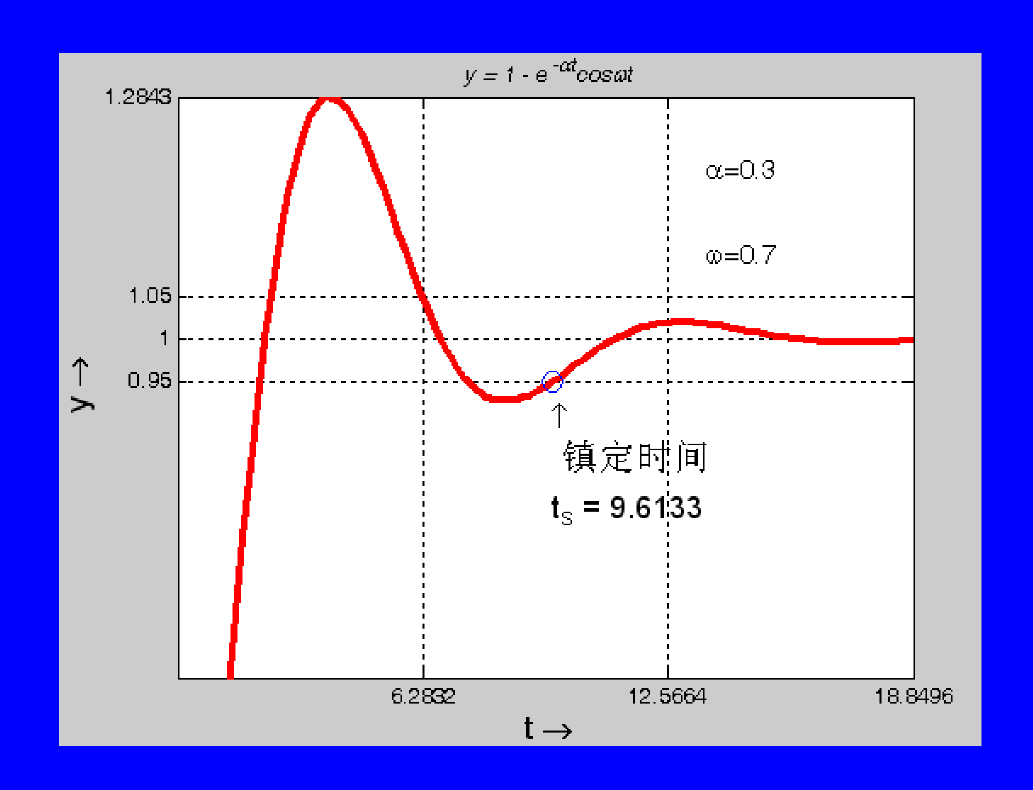


函数名	说明
gca	获得当前坐标轴对象的句柄
gcbf	获得当前正在执行调用的图形对象句柄
gcbo	获得当前正在执行调用的对象的句柄
gcf	获得当前图形对象的句柄
gco	获得当前对象的句柄
copyobj	复制图形对象及相应子对象
delete	删除图形对象
findobj	依属性值查找图形对象
get	获得对象属性
set	设置对象属性

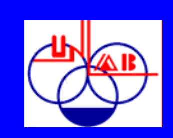
例题



绘制二阶系统阶跃响应曲线

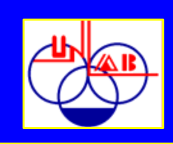


例题



```
clf;t=6*pi*(0:100)/100;y=1-exp(-0.3*t).*cos(0.7*t);
tt=t(find(abs(y-1)>0.05));                          %寻找大于0.05的元素
ts=max(tt);                                         %寻找tt中最大的元素
plot(t,y,'r-','LineWidth',3)
axis([-inf,6*pi,0.6,inf])
set(gca,'Xtick',[2*pi,4*pi,6*pi],'Ytick',[0.95,1,1.05,max(y)])
grid on
title('\it y = 1 - e^{ -\alphat}cos{\omegat}')
text(13.5,1.2,'\fontsize{12}{\alpha}=0.3')
text(13.5,1.1,'\fontsize{12}{\omega}=0.7')
hold on;plot(ts,0.95,'bo','MarkerSize',10);hold off
cell_string{1}='\fontsize{12}\uparrow';
cell_string{2}='\fontsize{16} \fontname{隶书}镇定时间';
cell_string{3}='\fontsize{6}  ';
cell_string{4}=['\fontsize{14}\rmt_{s} = ' num2str(ts)];
text(ts,0.85,cell_string)
xlabel('\fontsize{14} \bft \rightarrow')
ylabel('\fontsize{14} \bfy \rightarrow')
```

多次叠绘、双纵坐标和多子图



1)多次叠绘:

hold on，使当前轴及图形保持而不被刷新；hold off，不保持当前轴及图形。

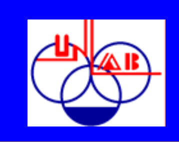
2) 双坐标图:

plotyy(X1,Y1,X2,Y2)　　以左右不同纵轴分别绘制X1-Y1、X2-Y2两条曲线
plotyy(X1,Y1,X2,Y2,Fun)　以左右不同纵轴把X1-Y1、X2-Y2绘制成Fun指定的形式的两条曲线

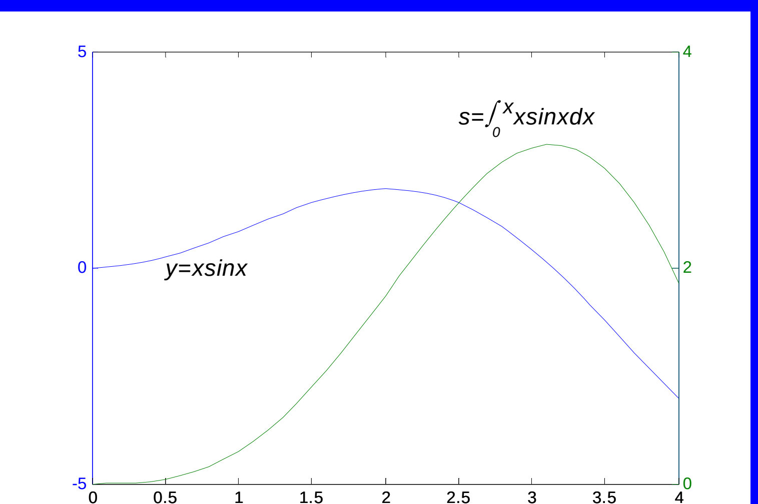
3)多子图:

采用subplot(m,n,k)使（m×n）幅子图中的第k个成为当前子图，再采用其它的图形绘制指令则可将图形绘制到指定的子图中。子图序号的编制原则是：左上方为第1幅，向右向下依次增大。

双坐标曲线绘制方法

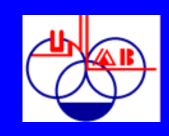


画出函数 $y = x\sin x$ 和积分 $s = \int_0^x (x\sin x)dx$ 在区间 $[0,4]$ 上的曲线



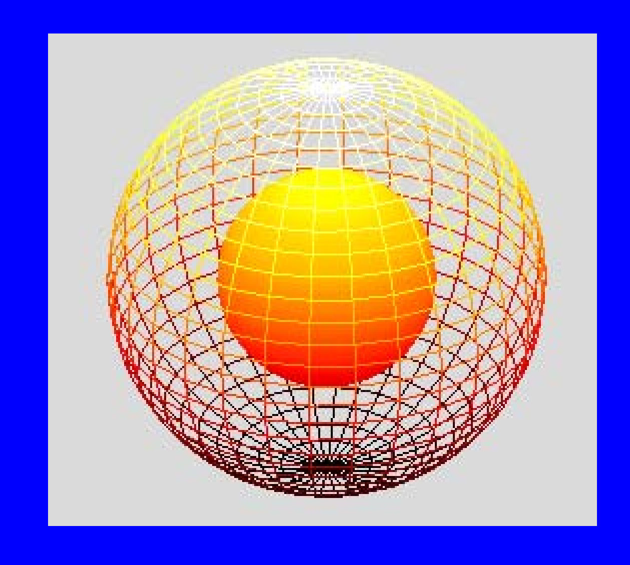

```
clf;
dx=0.1;x=0:dx:4;y=x.*sin(x);s=cumtrapz(y)*dx;        %梯形法求累计积分
plotyy(x,y,x,s),text(0.5,0,'\fontsize{14}\ity=xsinx')
sint='{\fontsize{16}\int_{\fontsize{8}0}^{  x}}';
text(2.5,3.5,['\fontsize{14}\its=',sint,'\fontsize{14}\itxsinxdx'])
```

Matlab三维图形



1) 三维曲线绘制命令 plot3
2) 三维网格图形绘制命令 mesh
3) 三维曲面绘制名利 surf

```
[X0 Y0 Z0]=sphere(30);
X=2*X0;Y=2*Y0;Z=2*Z0;
surf(X0,Y0,Z0);
shading interp
hold on
mesh(X,Y,Z)
colormap(hot)
hidden off
hold off
axis equal
axis off
```



Matlab图形绘制函数



Matlab图形绘制函数分属于以下帮助主题

- graph2d
- graph3d
- specgraph

程序的组成



Script文件或函数文件

数据输入 →(变量) 运算 →(变量) 数据输出

数据输入	运算	数据输出
键盘输入	数学运算	屏幕输出
文件输入	关系运算	文件输出
	逻辑运算	图形输出
	流程控制	

MATLAB关系运算符



关系运算符:

运　算	符　号	运　算	符　号
大于	>	小于	<
等于	==	不等于	~=
大于等于	>=	小于等于	<=

MATLAB逻辑、关系运算的规定



- 逻辑和关系运算均可以在任意具有相同维数的数组之间进行
- 标量可以和任意维数的数组运算
- 在所有关系表达式和逻辑表达式中，作为输入的任何非0数均被看作是“逻辑真”，只有0是“逻辑假”
- 所有关系表达式和逻辑表达式的运算结果是一个由0和1组成的逻辑数组
- 逻辑数组是“数值数组”的子类，可以按数值数组实施操作；它又有不同与普通数值数组的特性，即表示着对对象判断的真假；可用于数组寻访等特殊场合

MATLAB逻辑运算符



逻辑运算符:

<table>
<tr><th>运　算</th><th>符　号</th><th>运　算</th><th>符　号</th></tr>
<tr><td>与</td><td>&</td><td>或</td><td>|</td></tr>
<tr><td>非</td><td>~</td><td>异或</td><td>xor</td></tr>
</table>

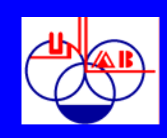
A=[0 1 1 0];　　　　B=[1 1 0 0]

A&B=[0 1 0 0]　A|B=[1 1 1 0]　~ A=[1 0 0 1]　xor(A,B)=[1 0 1 0]

先决逻辑运算符:

<table>
<tr><td>先决与</td><td>&&</td><td>先决或</td><td>||</td></tr>
</table>

逻辑、关系运算示例



```
A=-3:3;
B=3:-1:-3;
L1=~(A>0)
L2=~A>0
L3=~A
L4=A>-2&A<1
L5=A==B&L1
```

运算优先级:

1) 括号
2) 逻辑否
3) 乘除
4) 加减
5) 关系运算
6) 逻辑运算
7) 先决与
8) 先决否

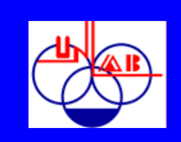
高

低

运行结果:

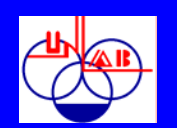
```
A =    -3    -2    -1     0     1     2     3
B =     3     2     1     0    -1    -2    -3
L1 =    1     1     1     1     0     0     0
L2 =    0     0     0     1     0     0     0
L3 =    0     0     0     1     0     0     0
L4 =    0     0     1     1     0     0     0
L5 =    0     0     0     1     0     0     0
```

MATLAB关系运算函数



isempty	数组是否为空
isequal	两个数组是否相等
any	数组有非零元素则结果为1
all	数组元素全非零则结果为1
find	数组非零元素的下标
isscalar	是否为标量
isvector	是否为向量
isnan	是否为非数
isinf	是否为无穷
isfinite	是否为有限

for循环结构



for 循环变量 = 表达式1（初值）：表达式2（步长）：表达式3（终值）
statements （语句组）
end字符串：包含在一对单引号中的字符集合

```
for i=1:10
x(i)=i;
end
x
```

- 为了得到高效代码，应尽量提高代码的向量化程度，避免使用循环结构
- 为了得到高效代码，在循环指令之前应尽量对数组进行预定义

while循环结构

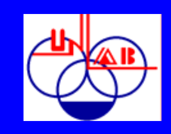


while cndition（表达式）
    statements（执行语句组）
end

Fibonacci数组的元素满足Fibonacci规则：$a_{k+2}=a_k+a_{k+1}$，（k=1,2,...);且a1=a2=1。求该数组中第一个大于10000的元素。

```
a(1)=1;a(2)=1;i=2;
while a(i)<=10000
    a(i+1)=a(i-1)+a(i);
    i=i+1;
end
i, a(i)
```

if-else-end分支结构



if语句的一般格式:

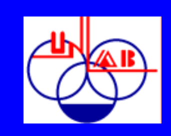
```
if condition1
     statements     %如果condition1的值为True，则执行该语句组
elseif condition2
    statements     %如果condition2的值为True，则执行该语句组
else
    statements     %如果condition1和condition2的都为False，则执行该语句组
end
```

用for循环指令寻求Fibonacci数组中第一个大于10000的元素

```
n=100;a=1:100;
for i=3:n
    a(i)=a(i-1)+a(i-2);
    if a(i)>=10000
        a(i),
        break;
    end
end
i
```

可以用break语句强制终止循环的运行

switch-case结构

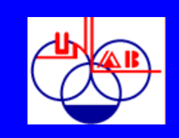


switch-case的一般格式:

```
switch test_expr          %测试表达式test_expr可以是标量或字符串
case value
    statements        % 当test_expr值是value时，执行该语句组
case {value1,value2,…}
    statements        % 当test_expr值是value1或value2或……时，执行该语句组
otherwise,
    statements
end
```

switch…case…otherwise语句的能力与if…else…end语句类似，但对多重选择的情况switch语句使代码更加易读。

try-catch结构



try-catch的一般格式：

```
try
        statements    % 此组语句总被执行。若正确则跳出此结构
catch
        statements    % 当上组语句出现执行错误后，该组语句被执行

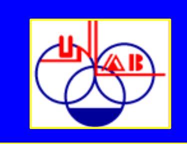
end
```

- 当两组语句都出错后，Matlab将跳出该结构
- 可以采用lasterr函数查询出错原因

```
N=4;A=[1 2 3];
try
        A_N=A(N)
catch
        A_N=A(end)
end
lasterr
```

```
A_N =     3
ans =
Index exceeds matrix dimensions.
```

本讲小结

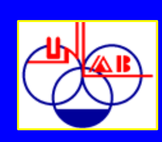


1) Matlab的基本数学运算符和运算函数的使用
   - 注意区别矩阵和数组的乘、除、乘方运算
2) Matlab数据输入输出功能，尤其是绘图功能的实现
3) Matlab函数文件的基本形式及其调用
4) 字符、单元数组、结构体的定义
5) Matlab的流程控制语句

第二讲
非线性方程（组）求解与迭代法

隋志军
化工学院软件应用教科组
2006-10

本章知识要点



- 数值计算
  - 单个非线性方程求解
  - 非线性方程组求解
  - 迭代法

- MATLAB
  - 求解非线性方程（组）的相关函数

本章的所要解决的典型问题



在945.36kPa（9.33atm）、300.2K时，容器中充以2mol氮气，试求容器体积。

已知此状态下氮气的P-V-T关系符合范德华方程，其范德华常数为a = 4.17atm•L/mol2，b = 0.0371L/mol

数学模型：

$$f(V)=(p+\frac{an^2}{V^2})(V-nb)-nRT=0$$

非线性方程（组）在化学工程中的作用



- 多组分混合溶液的沸点、饱和蒸气压计算
- 流体在管道中阻力计算
- 多组分多平衡级分离操作模拟计算
- 平衡常数法求解化学平衡问题
- 定态操作的全混流反应器的操作分析

求解非线性方程的方法

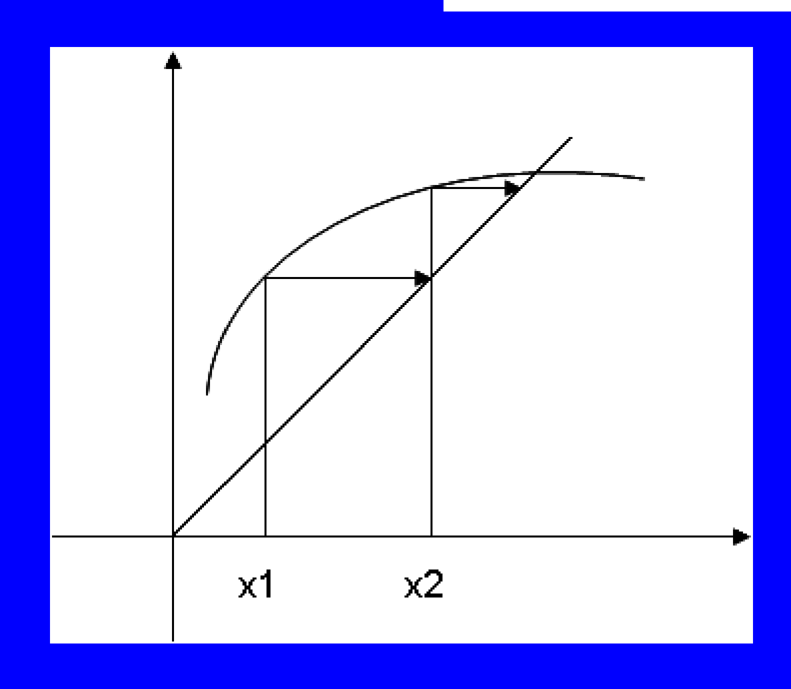


- 二分法
- 不动点迭代
- 威格斯坦法迭代
- 牛顿法
- 割线法

迭代法原理



不动点迭代： $f(x)=0 \rightarrow x=\varphi(x)$



迭代法意义示意图

已知：$x^3-x-1=0$

的解为1.324，以1.5为初值，采用以下两种迭代格式计算，结果如何？

$x_{k+1}=x_k^3-1$　　$x_{k+1}=(x+1)^{1/3}$

迭代法的收敛性



x1 x3 x2

$-1 < \phi'(x) < 0$

x3 x2 x1

$\phi'(x) > 1$

x2 x1 x3

$\phi'(x) < -1$

迭代法意义示意图



Wegstein法意义示意图

牛顿法意义示意图

割线法意义示意图

多项式函数



多项式： $P(x) = a_o x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n$

Matlab表达多项式方法： $P = [a_0, a_1, \cdots, a_{n-1}, a_n]$

Matlab多项式函数

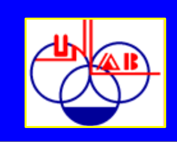
- Polyval 多项式的值
- Polyder 多项式微分
- Polyfit 多项式拟合
- Conv， Deconv 多项式乘法和除法
- Roots 多项式求根

简单多项式函数的使用




```
a=[1 11 55 125];
b=[1 1;1 1];
c=polyval(a,b)          % 求多项式a在b处的值
q=[2,3,5];
aq=conv(a,q)            % 多项式乘法
d=poly2sym(aq)          % 多项式向量表示为符号多项式
e=deconv(aq,a)          % 多项式除法
```

多项式求根函数roots



求解方程:

$$2x^4 - 5x^3 + 6x^2 - x + 9 = 0$$

```
p=[2 -5 6 -1 9];
sol=roots(p)
```

结果:

```
sol =

   1.6024 + 1.2709i
   1.6024 - 1.2709i
  -0.3524 + 0.9755i
  -0.3524 - 0.9755i
```

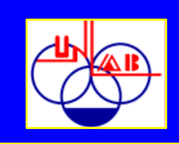
$$f(V) = (p + \frac{an^2}{V^2})(V - nb) - nRT = 0$$

```
P = 9.33;
T = 300.2;
n = 2;
a = 4.17;
b = 0.0371;
R = 0.08206
coef=[P, -(P*n*b+n*R*T), a*n^2 , -
    a*b*n^3];
V=roots(coef)
```

结果:

```
V =5.0028      0.2429      0.1092
```

非线性方程求解函数fzero



调用格式:

[x,fval,exitflag,output] = fzero(fun,x0,options, p1, p2, ...)

此函数的作用求函数fun在x0附件的零值点，x0是标量。

fval　　　　函数在解x处的值

exitflag　　　　程序结束情况，>0，程序收敛于解;
<0，程序没有收敛；＝0，计算达到了最大次数

output　　　　一个结构体，提供程序运行的信息;

output.iterations，迭代次数;

output.functions，函数fun的计算次数;

output.algorithm，使用的算法

options　　　　选项，可用optimset函数设定选项的新值

fun可以是函数句柄或匿名函数。

fzero函数的使用



1) sinx在3附近的零点

```
fzero(@sin,3)
```

2) cosx在[1,2]范围内的零点

```
fzero(@cos,[1,2])
```

3) $x^3 - 2\sin x = 0$

```
fzero(@(x) x^3-2*sin(x),1)
```

或者:

```
function Cha2demo1
x=fzero(@fun,1)
function y=fun(x)
y=x^3-2*sin(x);
```

4) $x^3 - 2x - 5 = 0$

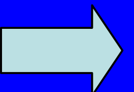
```
fzero(@(x) x^3-2*x-5,1);
roots([1 0 -2 -5])
```

fzero函数初值的选取



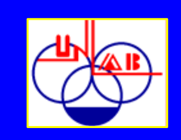
$$f(V) = (p + \frac{an^2}{V^2})(V - nb) - nRT = 0$$

```
P = 9.33;
T = 300.2;
n = 2;
a = 4.17;
b = 0.0371;
R = 0.08206;
V=fzero(@(V) P*V^3-
    (P*n*b+n*R*T)*V^2+ a*n^2*V -
    a*b*n^3,0)
```



V=0.1092

fzero函数初值的选取



$f(t) = (\sin^2 t)e^{-at} - b|t|$ 的零点

以t为自变量，取值范围为-10<t<10，a,b为参数，本例取值分别为0.1，0.5

```
function Cha2demo2
a=0.1;b=0.5;t=-10:0.01:10;
Y=sin(t).^2.*exp(-a*t)-b*abs(t);
clf,plot(t,Y,'r');hold on,plot(t,zeros(size(t)),'k');
xlabel('t');ylabel('y(t)'),hold off
zoom on
n=input('How many zero points are there?');
[tt,yy]=ginput(n);zoom off
for i=1:n
    [t0(i),y(i),exitflag]=fzero(@(t) sin(t)^2*exp(-a*t)-b*abs(t),tt(i));
end
disp('The zero points are:')
fprintf('%.4f\t',t0)
fprintf('\n')
```

fzero函数初值的选取



在945.36kPa（9.33atm）、300.2K时，容器中充以2mol氮气，试求容器体积。

已知此状态下氮气的P-V-T关系符合范德华方程，其范德华常数为a = 4.17atm•L/mol2，b = 0.0371L/mol

```
function PVT
P = 9.33;   % atm
T = 300.2;  % K
n = 2;      % mol
a = 4.17;
b = 0.0371;
R = 0.08206;
V0 = n*R*T/P
[V,fval] = fzero(@PVTeq,V0,[],P,T,n,a,b,R)
 % ------------------------------------------------------------
function f = PVTeq(V,P,T,n,a,b,R)
f = (P + a*n^2/V^2) * (V-n*b) - n*R*T;
```

fzero函数初值对结果的影响



预热到 T_0 的含有反应物的溶液原料，以一定的流量 Q，加入到容积为 V_R 的搅拌槽反应器中进行绝热反应。反应混合物连续排出。A 的进、出口浓度分别为 C_{A0} 和 C_A。反应溶液的密度为 ρ，比热容为 Cp。槽内及出口温度为 T，反应速度为：

$-r = kC_A^2$，式中 $k = k_0 \exp(-\frac{E}{RT})$。已知数据：$T_0$=450K，$C_{A0}(-H_r)/\rho C_p = 250\text{K}$，

E/R=10000K，$k_0 C_{A0} = e^{20}$，$\tau = V_R / Q = 0.25h$，试求反应转化率。

模型：由物料衡算和热量衡算可以获得模型方程如下

$$k_0 C_{A0}(1-x)^2 \tau \exp(-\frac{E}{RT}) - x = 0$$ （物料衡算式）

$$T - T_0 = \frac{(-\Delta H) C_{A0} x}{\rho C_p}$$ （热量衡算式）

```
function Conv=Cha2demo4
T0 = 450;
x0 = [0:0.1:1.0];
n=length(x0);
for i=1:n
   x(i)= fzero(@NonlinEq,x0(i),[],T0);
end
disp('The conversion could be')
fprintf('%.4f\t',x)
fprintf('\n')
% ----------------------------------------------------------------------
function f = NonlinEq(x,T0)
T = T0 + 250*x;
f = 0.25*(1-x)^2*exp(20-10000/T) - x;
```

结果:

```
The conversion could be
0.0408   0.0408   0.2804   0.2804   0.2804   0.2804
         0.8363   0.8363   0.8363   0.8363   1.0951
```

fsolve函数



- 与fzero函数只能求解单个方程的根不同，fsolve函数可求解非线性方程组的解。其算法采用的是最小二乘法。
- 调用格式:

  [x,fval,exitflag,output,jacobian] = fsolve(fun,x0,options, p1, p2, ...)
- 输入输入变量的意义同fzero函数。输出变量中的jacobian为函数fun在x处的Jacobian矩阵。

fsolve函数的使用



$$\begin{cases} \sin x + y^2 + \ln z = 0 \\ 3x + 2^y - z^3 + 1 = 0 \\ x + y + z = 5 \end{cases}$$

```
function Cha2demo6
x0=[1 1 1];
x=fsolve(@fun,x0)
function y=fun(x)
y(1)=sin(x(1))+x(2)^2+log(x(3))-7;
y(2)=3*x(1)+2^x(2)-x(3)^3+1;
y(3)=x(1)+x(2)+x(3)-5;
```

解得结果如下：x＝0.5991　2.3959　2.0050

fsolve函数的应用



在铜管内在**1 atm**下将异丙醇加热到**400**℃。已知铜是生产丙酮和丙醛的催化剂，或许还有某些异丙醇异构化为正丙醇。这三种产物的生成可用如下三个独立反应表示：

***i*C3H7OH(IP) = *n* C3H7OH(NP)　　　*K*1 = 0.064**

***i*C3H7OH(IP) = (CH3)CO(AC)+H2　　　*K*2 = 0.076**

***i*C3H7OH(IP) = C２H５ＣHO(PR) +H2　　　*K*3 = 0.00012**

后续加工步骤需要正丙醇，虽然可含丙酮，但丙醛含量不能超过**0.05(mol)%**。在上述反应条件下，是否存在违反这种规定的可能性？

数学模型：各反应的化学平衡方程如下

$$\frac{x_1}{1-x_1-x_2-x_3}=0.064$$

$$\frac{x_2(x_2+x_3)}{(1-x_1-x_2-x_3)(1+x_2+x_3)}=0.076$$

$$\frac{x_3(x_2+x_3)}{(1-x_1-x_2-x_3)(1+x_2+x_3)}=0.00012$$



```
function Cha2demo7
x0 = [0.05 0.2 0.01];
x = fsolve(@EquiC3,x0);
CAC=x(3)/sum(x)
if CAC<0.05
    disp('The AC concentration could not be over 0.05%')
else
    disp('The AC concentration could be over 0.05%')
end
function f = EquiC3(x)
f1 = x(1)-0.064*(1-x(1)-x(2)-x(3));
f2 = x(2)*(x(2)+x(3))-0.076*(1-x(1)-x(2)-x(3))*(1+x(2)+x(3));
f3 = x(3)*(x(2)+x(3))-0.00012*(1-x(1)-x(2)-x(3))*(1+x(2)+x(3));
f = [f1 f2 f3];
```

结果：The AC concentration could not be over 0.05%

水平管道内幂定律流体的流动－非牛顿流体流动所需管径

一管道内幂定律流体的水平流动：$\rho = 964\text{kg}\cdot\text{m}^{-3}$，质量流量 $G=6.67\text{kg}\cdot\text{s}^{-1}$，管长 $L=10\text{m}$，$\varepsilon = 5\times10^{-6}\text{m}$，$\Delta p = 15\text{kPa}$，$k = 1.8$，$n = 0.64$，$K = 1.48\text{N}\cdot\text{s}^{2-n}\cdot\text{m}^{-2}$。试用普遍化的雷诺数法估算此非牛顿流体流动情况所需的管道直径。

数学模型：

普遍化的雷诺数法定义：$\text{Re}_{gen} = \dfrac{D^{n'}u^{2-n'}\rho}{8^{n'-1}K'}$，式中，n'和 K'是剪切速率的函数。

对于幂定律流体：n＝n'，$K' = K\left[\dfrac{3n+1}{4n}\right]^{n}$

普遍化的 Re-f 关系式：

层流时（$\text{Re}_{gen} \le 2100$）　$f = \dfrac{16}{\text{Re}_{gen}}$

湍流时（$\text{Re}_{gen} > 2100$）　$\dfrac{1}{\sqrt{f}} = \dfrac{4.0}{n'^{0.75}}\lg(\text{Re}_{gen} f^{1-n'/2}) - \dfrac{0.4}{n'^{1.2}}$

管道直径 D 的计算是迭代过程，其起始估计值按下式估算：

$$D^{(1)} = \left[\frac{32K'L8^{n'-1}}{\Delta p}\left(\frac{4G}{\pi\rho}\right)^{n'}\right]^{1/(3n'+1)}$$（层流，k＝0）

D 的改进估计值：$D^{(r+1)} = \left[\dfrac{2f(L+Le)}{\rho\Delta p}\left(\dfrac{4G}{\pi}\right)^{2}\right]^{0.2}$，其中 $Le = \dfrac{kD}{4f}$

```
function Cha2demo5
rho = 961;G = 6.67;L = 10;eps = 5e-6;deltaP = 15e3;k = 1.8;n = 0.64;K = 1.48;n1 = n;
K1 = K*((3*n+1)/(4*n))^n;
 D1 = (32*K1*L*8^(n1-1)*(4*G/pi/rho)^n1)^(1/(3*n1+1));
D2 = 2*D1;
delta = 1e-4;
while abs((D2 - D1)/D2) > delta
    u = (G/rho)/(pi*D1^2/4);
    Regen = D1^n1*u^(2-n1)*rho/(8^(n1-1)*K1);
    if Regen <= 2100
                  f = 16/Regen;
   end
    if Regen > 2100
        f = FrictFactor(Regen,n1);
    end
    Le = k*D1/(4*f);
    D1 = D2;
    D2 = (2*f*(L+Le)/(rho*deltaP)*(4*G/pi)^2)^0.2;
end
fprintf('\t管道直径为D = %.4f %s\n',D2,'m')
fprintf('\t摩擦因子为f = %.4f %s',f,'m')
 % ------------------------------------------------------------
function f = FrictFactor(Regen,n1)
f0 = 16/Regen;      %4.53
f = fzero(@fFunc,f0,[],Regen,n1)
 % ------------------------------------------------------------
function F = fFunc(f)
F = 1/sqrt(f) - 4.0*log(Regen*f^(1-n1/2))/n1^0.75 + 0.4/n1^1.2;%4.54
```

本讲小结



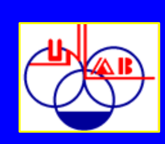
- ✓ sol=roots(C)
- ✓ [sol,feval,exitflag,output]=fzero(@fun,x0,options,p1,p2,...)
- ✓ [sol,feval,exitflag,output,jacobian]=fsolve(@fun,x0,options,p1,p2,...)

第三讲
矩阵操作与线性方程组求解

隋志军
化工学院软件应用教科组

2006-10

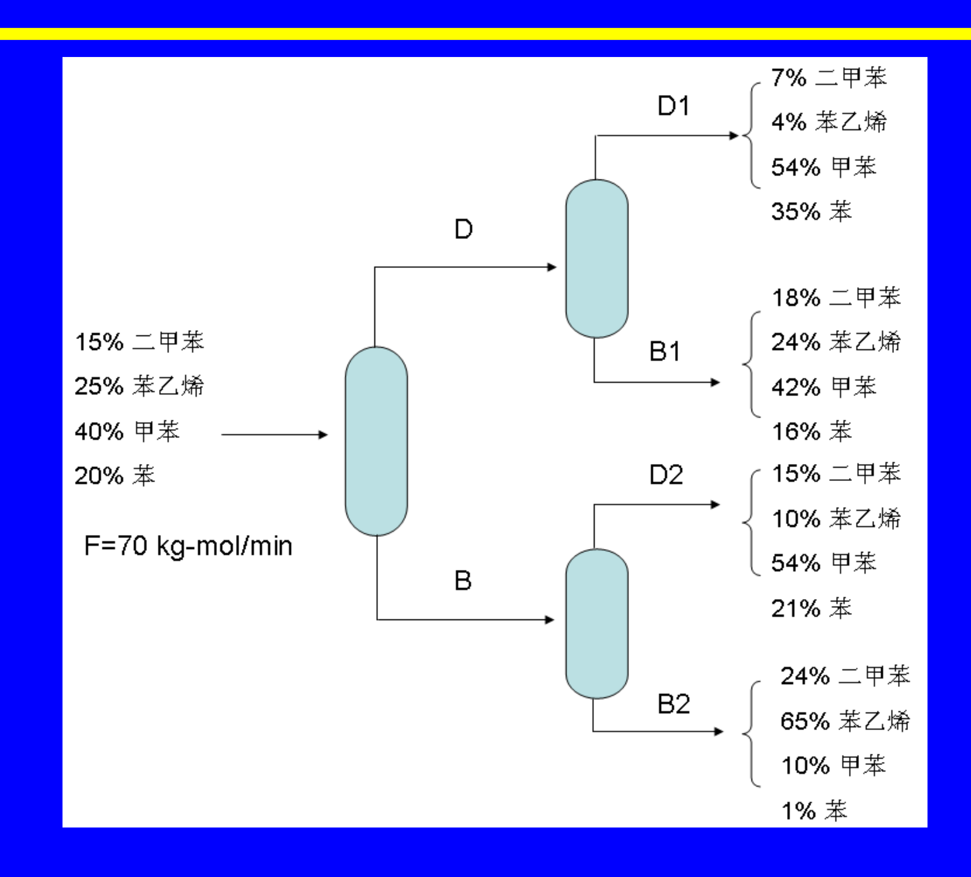
本章知识要点



- 数值计算
  - 线性方程组的直接解法
  - 线性方程组的迭代解法
- MATLAB
  - 矩阵的生成与操作
  - 矩阵运算函数

本章的所要解决的典型问题

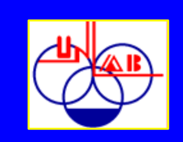




数学模型：

$$\sum(F(in)\cdot x_i(in)) = \sum(F(out)\cdot x_i(out))$$

线性方程组在化工模型中的作用



- 多组分体系的物料衡算
- 各种化合物的物理化学性质
- 稳态动力学计算
- 微分方程的差分方程求解

病态线性方程组



$$\begin{cases} 2x+6y=8 \\ 2x+6.00001y=8.00001 \end{cases}$$

x=1,y=1

$$\begin{cases} 2x+6y=8 \\ 2x+5.99999y=8.00002 \end{cases}$$

x=10,y=-2

$$\begin{cases} 0.2161x+0.1441y=0.1440 \\ 1.2969x+0.8648y=0.8642 \end{cases}$$

x=2,y=-2

$$\begin{cases} 0.2161x+0.1441y=0.144000001 \\ 1.2969x+0.8648y=0.86419999 \end{cases}$$

x=0.991,y=-0.487

线性方程组的求解方法



直接求解方法:

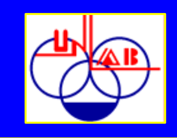
- 高斯消元法
- 高斯—约当消去法
- 追赶法

迭代解法:

- 雅可比（Jacobian）迭代
- 高斯-赛德尔（Gauss- Siedel）迭代
- 松弛（SOR）迭代
- 共轭梯度（CG）迭代

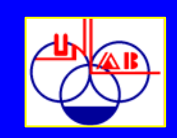
在Matlab线性方程组的解法归结为矩阵的除法命令

矩阵的生成



- 直接输入小矩阵
- 利用Array Editor
- 创建M文件输入大矩阵
- 利用矩阵操作命令
  - cat - 数组连接
  - reshape - 数组变维
  - repmat - 数组复制与平铺

常用矩阵生成函数



工具矩阵

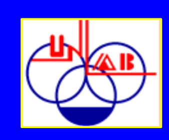
zeros - 全零阵
ones - 全一阵
eye - 单位阵
rand - 均匀分布随机阵
randn - 正态分布随机阵

B=zeros(n)
B=zeros(m,n)
B=zeros(size(A))

rand('state',n)
rand(N)

'state'
'twister'
'seed'

常用矩阵生成函数



特殊矩阵

compan	- Companion matrix.
gallery	- Higham test matrices.
hadamard	- Hadamard matrix.
hankel	- Hankel matrix.
hilb	- Hilbert matrix.
invhilb	- Inverse Hilbert matrix.
magic	- Magic square.
pascal	- Pascal matrix.
rosser	- Classic symmetric eigenvalue test problem.
toeplitz	- Toeplitz matrix.
vander	- Vandermonde matrix.
wilkinson	- Wilkinson's eigenvalue test matrix.

一维数组的寻访和赋值



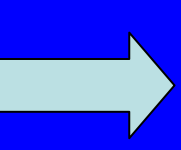
生成一（1×5）0状态下的均匀分布随机数组，并寻访1)数组的第三个元素；2) 数组的1，2，5个元素组成的子数组；3)前三个元素组成的子数组；4)寻访除前三个元素外的其它元素；5)前三个元素倒排形成的子数组；6)由大于0.5的元素构成的子数组；7)将元素的第1，4个元素赋值为0。

```
rand('state',0)
x=rand(1,5)
a1=x(3)
a2=x([1 2 5])
a3=x(1:3)
a4=x(3:end)
a5=x(3:-1:1)
a6=x(find(x>0.5))
x([1 4])=[0 0];x
```

矩阵的生成与寻访



```
A=[1 2 3;4 5 6;7 8 9];
A(5,5)=111
A(:,6)=222
AA=A(:,[1:6,1:6])
AA1=repmat(A,1,2)
AA2=repmat(A,2,2)
B=ones(2,6);AB_R=[A;B]
AB_C=[A,B(:,1:5)']
s=[1 3 6 8 9 11 14 16];
A(s)=0;A
```



```
A =[
    0     0     0     0     0   222
    2     5     8     0     0   222
    0     0     9     0     0   222
    0     0     0     0     0   222
    0     0     0     0   111   222
]
```

逻辑1标识



逻辑逻辑、关系运算

- any　　—若向量或数组的列向量任意元素不为0则返回真
- all　　—若向量或数组的列向量所有元素不为0则返回真
- find　　—寻找非零元素坐标

```
a=magic(5);
a (:,3)=zeros(5,1)
a 1=all(a(:,1)<10)
a 2=all(a>3)
a 3=any(a(:,1)>10)
a 4=any(a>10)
```

```
a=reshape([1:9],3,3);
a=1./a
f1=find(a)
f2=find(abs(a)>0.20&abs(a)<0.40)
```

二维数组的操作

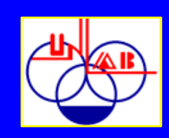


$$A=\begin{bmatrix} 2 & 6 & 5 & 22 & 2 \\ 1 & 6 & 3 & 8 & 94 \\ 13 & 4 & 5 & 7 & 21 \end{bmatrix}$$

求1)数组的第7个元素；2)绝对值最大的元素及其位置；3)所有绝对值大于10的元素

```
A=[2 6 5 22 2; 1 6 3 8 94; 13 4 5 7 21];
a1=A(7);
fprintf('\nThe 7th element is %2d\n',a1)
a2=max(max(A));
fprintf('\nThe maximum element of the matrix is
%2d\n',a2)
[R V]=find(A==a2);
fprintf('\nThe maximun element locates at %2dth Row and
%2dth colum\n',R,V)
L=abs(A)>10;
X=A(L)
```

矩阵的特殊操作



1)矩阵变维

- reshape(X,M,N)

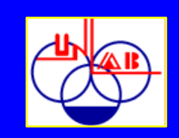
2)矩阵变向

- rot90（A），rot90（A,K）
- fliplr(A)，flipud(A)，flipdim(X,dim)

3)矩阵的抽取

- diag(X,K)
- tril(X)，tril(X,K)，triu(X)，triu(X,K)

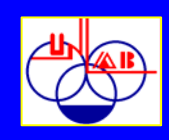
矩阵的基本性质



size	- Size of array.
length	- Length of vector.
ndims	- Number of dimensions.
numel	- Number of elements.
isempty	- True for empty array.
isequal	- True if arrays are numerically equal.
isequalwithequalnans	- True if arrays are numerically equal.

矩阵的基本生成、性质、操作命令归类于elmat主题，可通过>>help elmat 查看

Matlab矩阵函数



矩阵分析

- norm – 矩阵或向量范数;
- rank – 矩阵秩;
- det – 矩阵的行列式

线性方程组求解

- \和/ – 线性方程求解
- inv – 矩阵求逆
- cond – 矩阵条件数
- lu – LU分解

特征值和特征向量

- eig – 矩阵的特征值和特征向量
- svd – SVD分解

矩阵的分析等较为复杂的函数归类于matfun主题，可通过>>help matfun查看更为详细的信息

左除和右除的比较



用以下程序生成条件数很大的方程

```
randn('state',0);
A=gallery('randsvd',100,2e13,2);%产生条件数为2e13的100阶矩阵
x=ones(100,1);                  %生成真解
b=A*x;                          %用A和x生成b
```

比较左除法和求逆法的求解所用的时间和相对残差

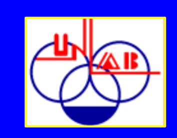
```
tic                %求逆法
xi=inv(A)*b;
ti=toc,eri=norm(x-xi),rei=norm(A*xi-b)/norm(b)
tic                %左除法
xd=A\b;
td=toc,erd=norm(x-xd),rei=norm(A*xd-b)/norm(b)
```

结果:

ti = 0.1950, eri = 0.0539, rei =0.0037

td =0.0156, erd =0.0779, rei =2.8535e-015

本章开始问题的求解



数学模型：

$$\begin{cases} 0.07D1 + 0.18B1 + 0.15D2 + 0.24B2 = 0.15 \times 70 \\ 0.04D1 + 0.24B1 + 0.10D2 + 0.65B2 = 0.25 \times 70 \\ 0.054D1 + 0.42B1 + 0.54D2 + 0.10B2 = 0.40 \times 70 \\ 0.035D1 + 0.16B1 + 0.21D2 + 0.01B2 = 0.20 \times 70 \end{cases}$$

求解程序：

```
A=[0.07 0.18 0.15 0.24;
 0.04 0.24 0.10 0.65;
 0.54 0.42 0.54 0.10;
 0.35 0.16 0.21 0.01];
B =[0.15*70; 0.25*70; 0.40*70; 0.20*70];
x =A\B
```

反应矩阵的秩与独立反应数



独立反应： 在一个存在多个反应的复杂反应体系中，某些反应的化学计量方程可以由其它反应的化学计量方程线性组合得到。如果m个同时发生的反应中，若每一个反应的计量方程都不能有其它反应的计量方程的线性组合得到，则称这m个反应是相互独立的。

反应矩阵：

$$4NH_3 + 5O_2 \rightarrow 4NO + 6H_2O$$
$$4NH_3 + 3O_2 \rightarrow 2N_2 + 6H_2O$$
$$4NH_3 + 6NO \rightarrow 5N_2 + 6H_2O$$
$$2NO + O_2 \rightarrow 2NO_2$$
$$2NO \rightarrow N_2 + O_2$$
$$N_2 + 2O_2 \rightarrow 2NO_2$$

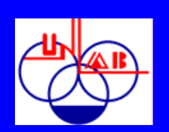
NH_3 O_2 NO H_2O N_2 NO_2

$$A = \begin{bmatrix} -4 & -5 & 4 & 6 & 0 & 0 \\ -4 & -3 & 0 & 6 & 2 & 0 \\ -4 & 0 & -6 & 6 & 5 & 0 \\ 0 & -1 & -2 & 0 & 0 & 2 \\ 0 & 1 & -2 & 0 & 1 & 0 \\ 0 & -2 & 0 & 0 & -1 & 2 \end{bmatrix}$$

b＝rank(A)

解得：**b＝3**，因此只有三个反应是独立的

线性方程组的迭代解法



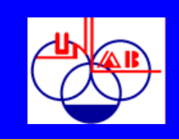
雅可比迭代：
$$x^{(k+1)} = D^{-1}(L+U)x^{(k)} + D^{-1}b$$

高斯-赛德尔迭代：
$$x^{(k+1)} = (D-L)^{-1}Ux^{(k)} + (D-L)^{-1}b$$

SOR迭代：
$$x^{(k+1)} = (D-\omega L)^{-1}[(1-\omega)D+\omega U]x^{(k)} + \omega(D-\omega L)^{-1}b$$

- 1<ω<2用于加速某收敛的迭代过程，而取0<ω<1用于非收敛迭代过程使其收敛。
- 最佳松弛因子的选取，一般需在计算过程中搜索寻优

线性方程组的迭代解法



在偏微分方程差分解法中出现如下典型方程组，试用分别用雅可比迭代，高斯-赛德尔迭代和SOR法求解。初值 $x_i^{(0)} = 10.0, (i = 1,2,\cdots,10)$

$$\left[\begin{array}{cccccc|c} -4 & 1 & & & & & -27 \\ 1 & -4 & 1 & & & & -15 \\ & 1 & -4 & 1 & & & -15 \\ & & \ddots & \ddots & \ddots & & \vdots \\ & & & \ddots & \ddots & 1 & \vdots \\ & & & & 1 & -4 & -15 \end{array}\right]$$

```
function Cha3demo7
 A=[-4 1 0 0 0 0;1 -4 1 0 0 0; 0 1 -4 1 0 0;0 0 1
-4 1 0; 0 0 0 1 -4 1;0 0 0 0 1 -4];
 D=diag(diag(A)); U=-triu(A,1); L=-tril(A,-1);
 x0=0*ones(1,6)';
 b=[-27 -15 -15 -15 -15 -15]';
 %Jacobian Iteration
 BJ=D\(L+U); gJ=D\b; y=BJ*x0+gJ;
 n=1;
 while norm(y-x0)>=1e-6
     x0=y;
     y=BJ*x0+gJ;
     n=n+1;
 end
disp('The solution of Jacobian iteration are')
fprintf('\nn=%3d\n',n)
disp('y='),disp(y')
```

```
%Gauss-Seidel
BG=(D-L)\U;gG=(D-L)\b;y=BG*x0+gG;
n=1;
 while norm(y-x0)>=1e-8
      x0=y;
      y=BG*x0+gG;
      n=n+1;
 end
disp('The solution of Gauss-Seidel iteration
are')
fprintf('\nn=%3d\n',n)
disp('y='),disp(y')
```

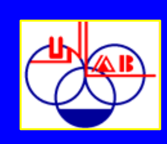
```
%SOR
omiga=1.08;BS=(D-omiga*L)\((1-
omiga)*D+omiga*U);gS=omiga*((D-omiga*L)\b);
y=BS*x0+gG;
n=1;
 while norm(y-x0)>=1e-8
     x0=y;
     y=BS*x0+gS;
     n=n+1;
 end
disp('The solution of SOR iteration are')
fprintf('\nn=%3d\n',n)
disp('y='),disp(y')
```

第四讲
插值、拟合与数值微分和积分

化工学院软件应用教科组

2006-10

本章知识要点



- 插值方法(interp,spline)
- 拟合方法(polyfit,csaps)
- 数值微分(polyder, fnder)
- 数值积分(quad, quadl, fnint)

插值、拟合、数值微分、数值积分在化工计算中的作用



- 表格式物性数据的内插
- 离散实验数据点的处理
- 状态方程计算流体的焓和熵
- 微分法反应动力学方程拟合
- 等温活塞流反应器的设计计算
- 微观离析反应器的计算

插值简介



插值的数学问题可以描述为：已知n+1个数对{xi, f(xi)}，其中i = 0,1…n，（xi互不相同，称之为节点），求取函数g(xi)=f(xi)。

当{xi, f(xi)}有相当的精确度，但它们的函数关系难以确定或难以计算时，则可利用这些数据点来构造一个较简单的函数来近似表达原函数关系。

根据逼近函数的不同，常见的插值方法：

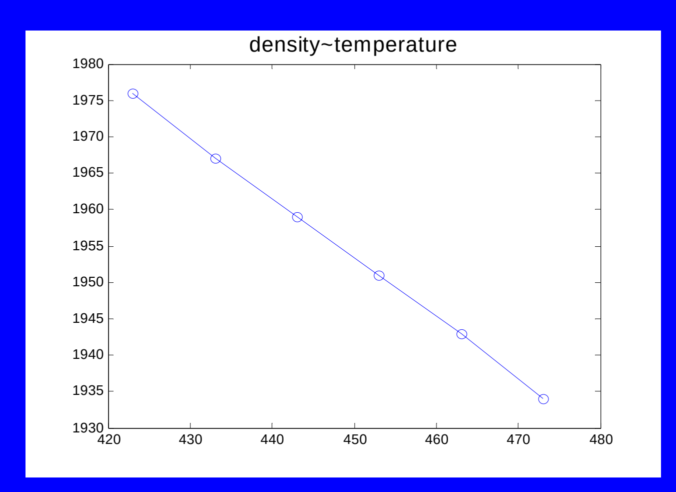
- **Lagrange**多项式插值（线性插值）
- 分段插值
- 三次样条插值
- 三角插值
- 有理式插值

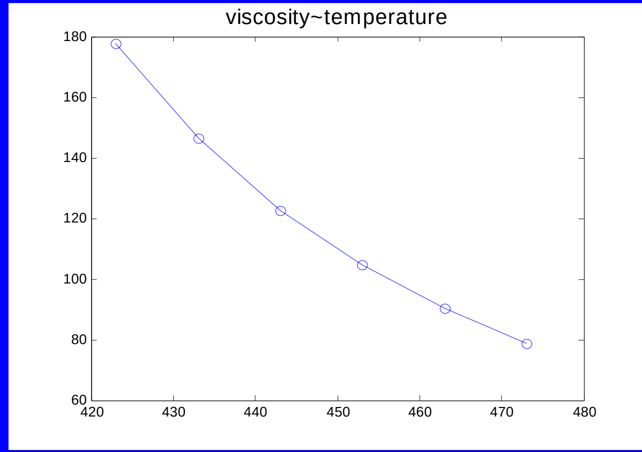
插值方法的选择



已知熔盐在423～473K的密度和粘度如下表所示，估计450K时的密度和粘度。

温度 (K)	密度 (kg/m³)	粘度 (Pa·S)
423	1976	177.58
433	1967	146.51
443	1959	122.79
453	1951	104.6
463	1943	90.26
473	1934	78.79





Matlab的插值（Interpolation）函数



插值方法	Matlab函数	插值方法	Matlab函数
一维插值	interp1	使用FFT方法的一维插值	interpft
快速一维插值	interpq	分段三次Hermite插值	pchip
二维插值	interp2	三次样条插值	spline
三维插值	interp3		
N维插值	interpn		

一维插值interp1



调用格式：

yi = interp1(x,y,xi) 已知数据向量（x,y），计算并返回在插值向量xi处的函数值

yi=interp1(x,y,xi, ‘method’)

yi=interp1(x,y,xi, ‘method’, ‘extrap’)

‘method’用于指定插值算法，其值可以是:

‘nearest’——最近插值

‘linear’——线性插值（默认值）

‘spline’——分段三次样条插值

‘pchip’——分段三次Hermite插值

‘cubic’——与‘pchip’相同

注意：

- ✓向量x为单调。若y为矩阵，则对y的每一列进行插值
- ✓向量xi中有元素不在x的范围内，则对应yi值为NaN
- ✓ ’extrap’用于指定当向量xi中有元素不在x的范围内时，采用’method’所指定的插值算法进行外插计算与之对应的yi值

一维插值方法比较

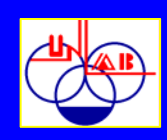


已知x=[0:10]，y=[0　0.8415　0.9093　0.1411　-0.7568　-0.9589　-0.2794　0.6570　0.9894　0.4121　-0.5440];(y = sinx)，比较一维线性、线性最近、立方和三次样条插值所得xi = 0,0.15,0.30,0.45,…,10处的值yi。如果初始数据点为x = 0,2,4,…,10，y = sinx，以上方法插值效果。

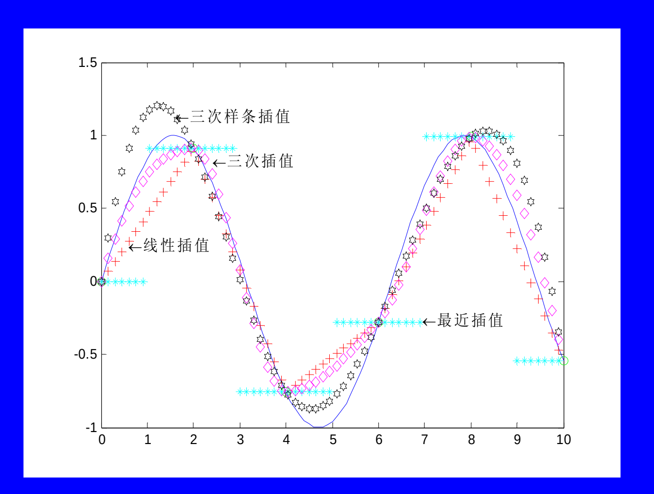
```
x=0:1:10;
y =[0  0.8415 0.9093 0.1411  -0.7568 -0.9589 -0.2794   0.657  0.9894   0.4121   -0.5440];
plot(x,y,'co'),hold on
fplot(@sin,[0 10])

xi=0:0.15:10;
yi=interp1(x,y,xi);
plot(xi,yi,'r+'),text(0.7028,0.4649,'线性插值\rightarrow')
yi2=interp1(x,y,xi,'nearst');
plot(xi,yi2,'c*'),text(3.537,0.1374,'\leftarrow最近插值')
yi3=interp1(x,y,xi,'cubic');
plot(xi,yi3,'md'),text(2.408,0.8333,'\leftarrow三次插值')
yi4=interp1(x,y,xi,'spline');
plot(xi,yi4,'kh'),text(4.62,0.8158,'三次样条插值\rightarrow')
```

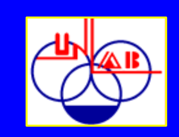
初始数据对于插值的影响



```
x=0:2:10;y=sin(x);
plot(x,y,'go'),hold on
ezplot(@sin,[0 10])
xi=0:0.15:10;
yi=interp1(x,y,xi);
plot(xi,yi,'r+'),text(0.5876,0.2537,'\leftarrow线性插值')
yi2=interp1(x,y,xi,'nearst');
plot(xi,yi2,'c*'),text(6.947,-0.258,'\leftarrow最近插值')
yi3=interp1(x,y,xi,‘pchip');
plot(xi,yi3,'md'),text(2.408,0.8333,'\leftarrow三次插值')
yi4=interp1(x,y,xi,'spline');
plot(xi,yi4,'kh'),text(1.601,1.138,'\leftarrow三次样条插值')
```



spline与pchip



Spline()的调用格式为:

yi=spline(x,y,xi)　此函数等同于yi=interp1(x,y,xi,'spline')

pp=spline(x,y)　返回三次样条插值的分段多项式形式的向量

spline函数可以保证插值函数的三阶导数连续

pchip()的调用格式为:

yi=pchip(x,y,xi)　此函数等同于yi=interp1(x,y,xi,'pchip')

pp=pchip(x,y)　返回三次样条插值的分段多项式形式的向量

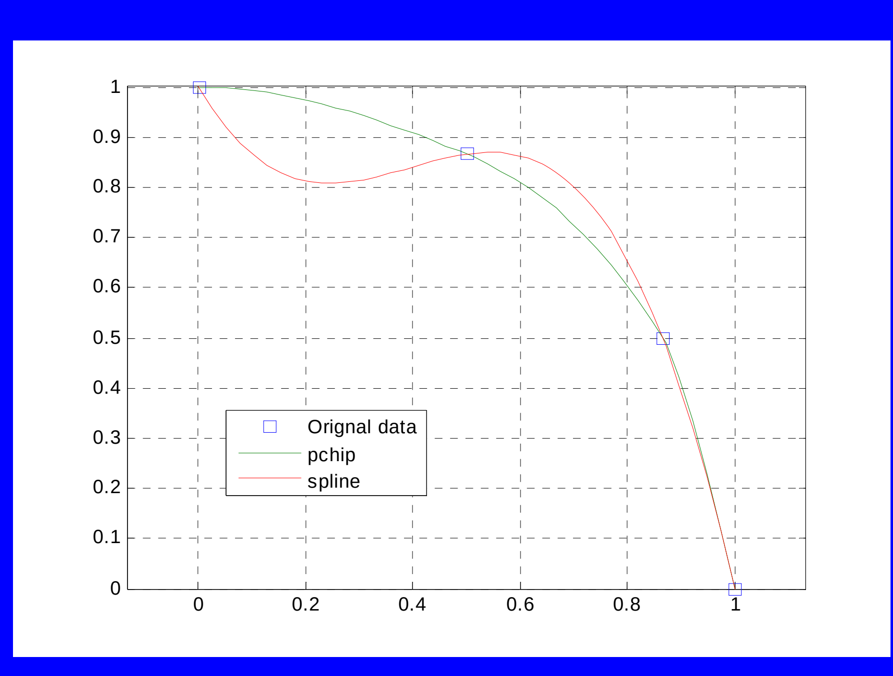
pchip与spline的区别



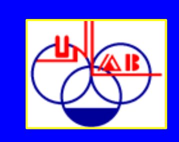
pchip与spline的区别：三次样条在相邻的节点上并不保证单调性；而Hermite分段三次样条则可保证插值的局部单调性

利用点(x=sin(kπ/6),y=cos(kπ/6)，其中k=[0 1 2 3]来逼近单位圆的前四分之一圆周。比较pchip与spline的差别。

```
t=linspace(0,pi/2,4)
x=cos(t);y=sin(t);
xx=linspace(0,1,40);
plot(x,y,'s',xx,[pchip(x,y,xx);spline(x,y,xx)])
grid on, axis equal
legend('Orignal data','pchip','spline')
```



其它样条插值函数－csape



csape()的调用格式为:

pp=csape(x,y)

pp=csape(x,y,conds)

conds可为以下字符串:

‘complete’	指定端点处一阶导数
‘second’	指定端点处二阶导数
‘periodic’	左右端点处一、二阶导数相等
‘not-a-knot’	第二个和倒数第二个节点处三阶导数连续
‘variational’	自然边界条件，即端点二阶导数为零

除csape外，Matlab的样条工具箱还提供了其它样条插值函数，如csapi，spapi等

csape的使用



设f(x)为区间[0,3]上的函数，剖分节点为xi=[0 1 2 3]；节点上的函数值为yi=[0 0.5 2 1.5]，左右端点处的一阶导数值分别为0.2和-1。试求区间[0,3]上满足上述条件的三次样条插值函数

程序：

```
x=[0 1 2 3];
y=[0.2 0 0.5 2.0 1.5 -1];
pp=csape(x,y,'complete')
```

几个有用的函数



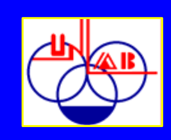
1. [breaks coff k m n]=unmkpp(pp)命令可以看到样条函数的具体信息

   其中coff是一个矩阵，其第i行是第i个三次多项式的系数，多项式形式如下:

$$s_i(x) = a_0(x - x_{i-1})^3 + a_1(x - x_{i-1})^2 + a_2(x - x_{i-1}) + a_3$$

2. fnval或ppval可以计算样条函数在指定点的函数值
3. fnder和fnint可分别计算样条函数的导数和积分
4. fnplt可用于绘制样条函数的图形

二维插值：interp2



调用格式：

zi=interp2(x,y,z,xi,yi,'method')

'method'算法属性值可以是;

'nearest'——最近插值

'linear'——线性插值（默认）

'spline'——三次样条插值(spline)

'cubic'——立方插值

二维插值函数的使用

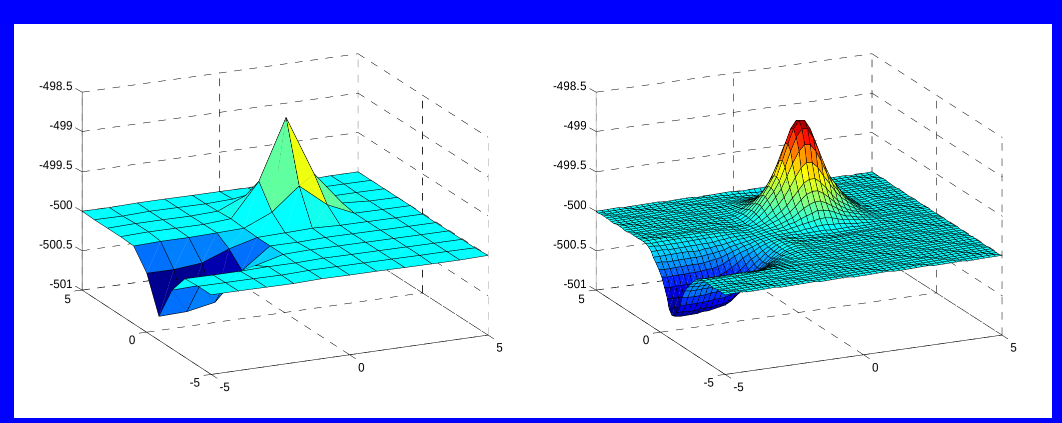


假设有一组分度系数的“海底深度测量数据”，由以下一段程序生成：

```
randn('state',2);
x=-5:5;y=-5:5;[X,Y]=meshgrid(x,y);
Z=-500+1.2*exp(-((X-1).^2+(Y-2).^2))-0.7*exp(-(exp(X+2).^2+(Y+1).^2));
surf(X,Y,Z)，view(-25,25)
```

试由插值方式绘制海底形状图。

```
xi=linspace(-5,5,50);
yi=linspace(-5,5,50);
[XI,YI]=meshgrid(xi,yi);
ZI=interp2(X,Y,Z,XI,YI,'*cubic');
surf(XI,YI,ZI)
view(-25,25)
```



拟合简介



- 拟合和插值的区别在于:

  1. 拟合时，所得函数不需要过所有数值点

  2. 插值函数不宜外推，拟合函数在某些情况则可以

- 拟合方法中最常用的是最小二乘曲线拟合
- 最小二乘法的基本思路是使拟合因变量y在给定点xi上使残差平方和最小
- 用于拟合的函数可以是多样的，本讲只介绍多项式函数拟合和样条函数拟合

最小二乘多项式拟合：polyfit



调用格式如下：

```
p=polyfit(x,y,n)
[p,s]=polyfit(x,y,n)
```

说明：

■输入参数：(x,y)为已知数据向量，n为多项式阶数；输出参数p为拟合生成的多项式的系数向量（长度为n+1），s为结构参数，供函数polyval()调用以获得误差估计值。

■函数polyval()常常与polyfit()联合使用，其调用格式为：y=polyval(p,x)。

■ [y,deltay]=polyval(p,xi,s), 返回xi处的拟合函数值，deltay是50%置信区间的y的误差。

多项式次数对拟合效果的影响



已知下表数据，用polyfit进行多项式拟合

x	0.5	1.0	1.5	2.0	2.5	3.0
y	1.75	2.45	3.81	4.80	8.00	8.60

```
x=0.5:0.5:3;y=[1.75 2.45 3.81 4.80 8.00 8.60];
xi=0.5:0.05:3;
[a2,S2]=polyfit(x,y,2);[y2,delta2]=polyval(a2,xi,S2);
plot(x,y,'ro',xi,y2,'m-'),text(1.04,4.67,'二次多项式拟合\leftarrow')
hold on,
[a4,S4]=polyfit(x,y,4);[y4,delta4]=polyval(a4,xi,S4);
plot(x,y,'ro',xi,y4,'b-'),hold on,text(1.563,6.775,'四次多项式拟合\leftarrow')
[a7,S7]=polyfit(x,y,7);[y7,delta7]=polyval(a7,xi,S7);
plot(x,y,'ro',xi,y7,'k-'),hold on,text(1.931,9.532,'七次多项式拟合\leftarrow'),hold off
```

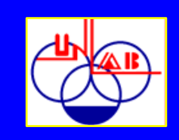
最小二乘法拟合生成样条曲线



函数名	曲线类型	拟合准则	是否平滑处理
csaps()	三次样条曲线	最小二乘法	是
spap2()	B样条曲线	最小二乘法	否
spaps()	B样条曲线	最小二乘法	是

- 拟合得到曲线函数sp以后，可利用fnval()计算任意自变量下的函数值。

函数csaps()的用法



功能：平滑生成三次样条函数，即对于数据(xi,yi)，所求的三次样条函数y=f(x)满足：

$$\min \; p \cdot \sum_i w_i (y_i - f(x_i))^2 + (1-p)\int \lambda(t)\left(D^2 f\right)t^2 dt$$

调用格式：sp=csaps(x,y,p)
ys=csaps(x,y,p,xx,w)

输入参数：x,y　要处理的离散数据(xi,yi)
p　平滑参数，取值区间为[0,1]。当p=0时，相当于最小二乘直线拟合；当p=1时，相当于“自然的”三次样条拟合
xx　用于指定在给定点xx上计算其三次样条函数值(ys)
w　权值（权重），默认为1

输出参数：sp　拟合得到的样条函数
ys　在给定点xx上的三次样条函数值

函数spap2 ()的用法



功能：用最小二乘法拟合生成B样条曲线，即对于数据(xi,yi)，所求k次样条函数y=f(x)满足：

$$\min \sum_i w_i \left(y_i - f(x_i)\right)^2$$

调用格式:	sp=spap2(knots,k,x,y) sp=spap2(knots,k,x,y,w)	
输入参数:	knots	节点序数(knot sequence)
	k	样条函数的阶次，一般取k=3，有时取k=4
	x,y	要处理的离散数据(xi,yi)
	w	权值（权重），默认为1
输出参数:	sp	拟合得到的样条函数

函数spaps ()的用法



功能：平滑生成B样条函数

$$\min \sum_i w_i \left(y_i - f(x_i)\right)^2$$

调用格式: sp=spaps(x,y,tol)

[sp,ys]=spaps(x,y,tol,m,w)

输入参数:	x,y	要处理的离散数据(xi,yi)
	tol	光滑时的允许精度
	m	默认值是2，即平滑生成三次B样条曲线
	w	权值（权重），默认为1
输出参数:	sp	拟合得到的样条函数
	ys	在x上经平滑处理的B样条函数值

函数csaps()的用法



采用样条拟合函数csaps进行拟合，比较p的取值对于拟合效果的影响

x	0.5	1.0	1.5	2.0	2.5	3.0
y	1.75	2.45	3.81	4.80	8.00	8.60

```
x=0.5:0.5:3;
y=[1.75 2.45 3.81 4.80 8.00 8.60];
xi=0.5:0.05:3;
plot(x,y,'ro')
hold on,
C=[0 0 0;1 0 0;0 0 1; 0 1 1; 0.5 0.5 0.5];
j=1;
for i=0:0.25:1.0
    Cj=C(j,:);
    ys(j,:)=csaps(x,y,i,xi);
    plot(xi,ys(j,:),'color',Cj),pause
    j=j+1;
end
```

对于微小的数据扰动，多项式拟合通常比样条函数更为敏感

数值微分



- 对于列表型函数往往需要用数值方法计算函数的微分
- 数值微分的基本方法
  1. 差分
  2. 利用插值（拟合）多项式求微分
  3. 利用三次样条插值（拟合）函数求微分
- 数值微分可以放大误差，应谨慎使用

Matlab数值微分实现方法



有限差分法：用差分函数diff()近似计算导数

多项式拟合方法：

离散数据 $\xrightarrow{polyfit()}$ 多项式拟合函数 $\xrightarrow{polyder()}$ 导函数pp $\xrightarrow{polyval()}$ 在xi的导数值

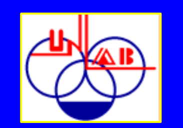
三次样条插值方法

离散数据 $\xrightarrow{spline()}$ 样条插值函数cs $\xrightarrow{fnder()}$ 导函数pp $\xrightarrow{fnval()}$ 在xi的导数值

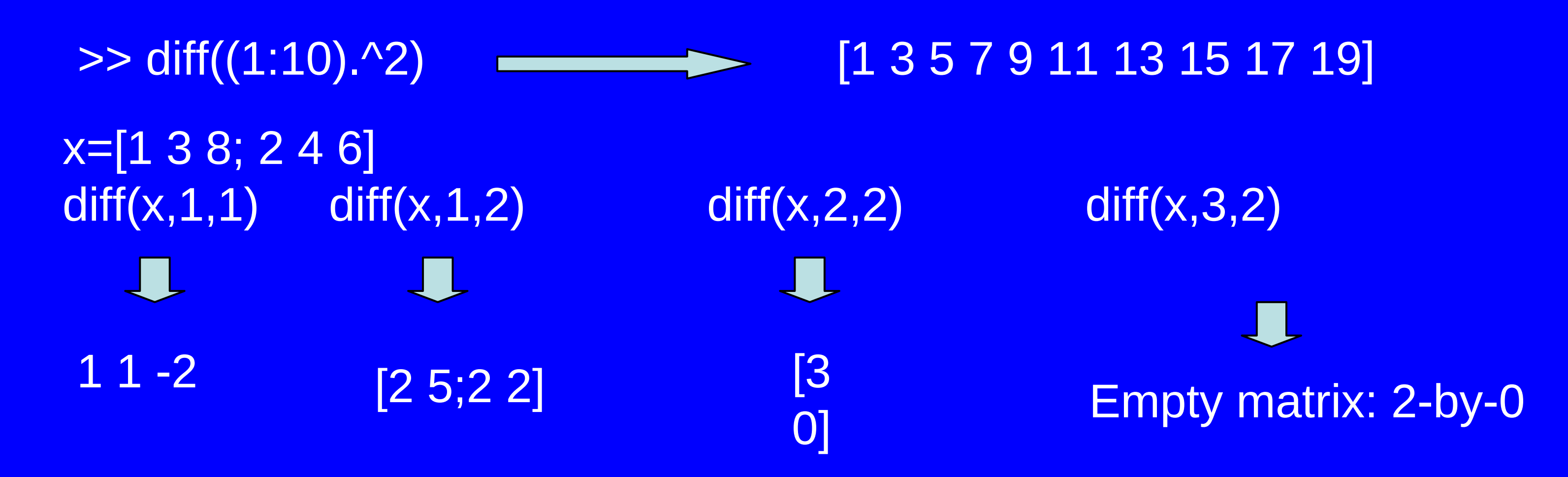
样条拟合方法

离散数据 $\xrightarrow{csaps()或spap2a()或spaps()}$ 样条拟合函数sp $\xrightarrow{fnder()}$ 导函数pp $\xrightarrow{fnval()}$ 在xi的导数值

函数diff



对于向量X，diff(X)表示了[X(2)-X(1) X(3)-X(2) ... X(n)-X(n-1)].
对于矩阵X，diff(X)表示了[X(2:n,:) - X(1:n-1,:)]
diff(x,n,dim)得到矩阵x在dim维上的n阶差值



利用diff函数求sin(x)在[0,10]上的导数值

```
fplot(@cos,[1 10])
hold on
h=1;x=1:h:10;
hh=0.01;xx=1:hh:10;
diffy=diff(sin(x))/h;
diffyy=diff(sin(xx))/hh;
plot([1:h:10-h],diffy,'r:',[1:hh:10-hh],diffyy,'k-o')
```

函数gradient



FX = gradient(F)
[FX,FY] = gradient(F)
[Fx,Fy,Fz] = gradient(F)
[...] = gradient(F,h)
[...] = gradient(F,h1,h2,...)

其中

1. Fx=dF/dx，FY=dF/dy，Fz=dF/dz; 当F为向量时，dF=gradient(F)是一维梯度
2. h1, h2，……是步长，缺省值为1
3. 如果h1，h2，……为向量,其长度必须匹配F的维数

例题



某一液相反应浓度随时间变化的实验数据如下表:

t/min	0	0.2	0.6	1.0	2.0	5.0	10.0
C(g/L)	5.19	3.77	2.30	1.57	0.8	0.25	0.094

试t=0.1,0.4min时的反应速度

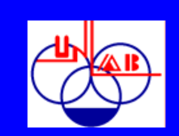
```
x=[0 0.2 0.6 1 2 5 10];
C=[5.19 3.77 2.30 1.57 0.8 0.25 0.094];
plot(x,C,'bo'),hold on
pp=csaps(x,C);fnplt(pp)
dC=fnder(pp);
dC1=ppval(dC,0.1)
dC2=fnval(dC,0.4)
fprintf('The reaction rate at 0.1min and 0.4min are%5.4f,%5.4f...
(g/(L min)respectively\n',dC1,dC2)
```

数值积分



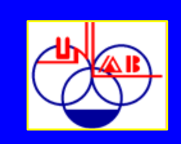
- 数值积分在数值计算中有着重要作用,许多数值计算问题可以转化为数值积分问题,如常微分方程初值问题等
- 通常可用逼近多项式Pn(x)来代替被积函数f(x)，计算积分$\int_a^b P_n(x)dx$
- 构造数值积分的方法很多，主要有Newton-Cotes系列数值积分法、Gauss积分法和Romberg积分法等

数值积分



MATLAB函数	公式
quad	自适应Simpson求积公式（低阶）
quadl	自适应Lobatto求积公式；精度高，最常用
trapz	梯形求积公式；速度快，精度差
cumtrapz	梯形法求一个区间上的积分曲线
cumsum	等宽距法求一个区间上的积分曲线，精度很差
fnint	利用样条函数求不定积分；与spline，ppval配合使用，主要应用于表格“函数”积分

梯形法数值积分：trapz()



调用格式：　z=trapz(y)

- 用梯形求积方法计算y的积分近似值。
- 对于向量y，trapz(y)返回y的积分;
- 对于矩阵y，trapz(y)返回一行向量，向量中的各元素为矩阵y的对应列 向量的积分值;

自适应Simpson法数值积分：quad()



调用格式: q=quad(@fun,a,b)
q=quad(@fun,a,b,tol)
q=quad(@fun,a,b,tol,trace,p1,p2,……)

输入参数:
- fun 被积函数。在定义fun时，被积函数表达式必须是向量形式，即表达式必须使用点运算符(.*、./和.^)以支持向量
- a,b 即积分限[a,b]
- tol 绝对误差限，默认值为1.e-6
- p1,p2…… 直接传递给函数fun的已知参数

输出参数:
- q 积分结果

自适应Lobatto法数值积分：**quadl()** 调用格式同quad

不同积分函数的比较

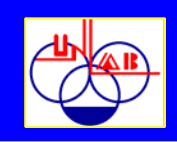


求积分: $\int_0^{3\pi} e^{-0.5t} \sin(t+\pi/6)dt$

比较cumsum，trapz，quad，quadl的积分精度，该积分的精确解为0.9008407878

```
function Cha4demo5
format long
d=pi/1000;
t=0:d:3*pi;
nt=length(t);
y=fun(t);
sc=cumsum(y)*d;
scf=sc(nt)
z=trapz(y)*d
qd=quad(@fun,0,3*pi,[],1)
qd2=quadl(@fun,0,3*pi,[],1)
EV=0.9008407878;
err(1)=abs(scf-EV);
err(2)=abs(z-EV)*10;
err(3)=abs(qd-EV)*10;
err(4)=abs(qd2-EV)*10;
bar(err),title('不同积分方法比较'),
colormap(summer)
text(0.7,7e-4,'矩形法','fontsize',20)
text(1.5,5e-4,'梯形法','fontsize',20)
text(2.4,3e-4,'Simpson法','fontsize',20)
text(3.4,1e-4,'Lobatto法','fontsize',20)
%-----------------------------------------------------------
function y=fun(t)
y=exp(-0.5*t).*sin(t+pi/6)
```

广义积分



1. 奇点积分

$$\int_0^1 \frac{dx}{\sqrt{x}(\exp(x)+1)}$$

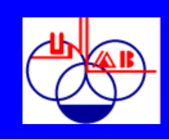
```
fun=inline('1./(sqrt(x).*(exp(x)+1))');
quadl(fun,0,1)
quadl(fun,eps,1)
```

2. 无穷积分

$$\int_0^{+\infty} \exp(\sin x - x^2/100)dx$$

可先选取一个有限的积分区间，如[0,100]计算；在选择一个较大的积分区间，如[0 200]计算，如两次计算结果的差满足一定的精度要求，则可认为此值即为无穷积分的值

多重数值积分



二重积分函数:

SS=dblquad(fun,xmin,xmax,ymin,ymax,tol,method)

三重积分函数:

SSS=triplequad(fun,xmin,xmax,ymin,ymax,zmin,zmax,tol,method)

说明:

Matlab只能处理积分限为常数的多重积分，对内积分上下限为外积分变量的积分问题,需自行编写函数求解。

样条函数在数值积分与微分中的应用



已知x=(0:0.1:1)*2*pi;y=sin(x)，求其积分和导数

```
x=(0:0.1:1)*2*pi;y=sin(x); pp=spline(x,y);
int_pp=fnint(pp);
der_pp=fnder(pp);
% 在基础区间上，计算三个样条函数与理论值的最大误差
xx=(0:0.01:1)*2*pi;
err_yy=max(abs(ppval(pp,xx)-sin(xx)))
err_int=max(abs(ppval(int_pp,xx)-(1-cos(xx))))
err_der=max(abs(ppval(der_pp,xx)-cos(xx)))
% 计算y(x)在区间[1,2]上的定积分
DefiniteIntegral.bySpline=ppval(int_pp,[1,2])*[-1;1];
DefiniteIntegral.byTheory=(1-cos(2))-(1-cos(1));
% 计算dy(3)/dx
Derivative.bySpline=fnval(der_pp,3);
Derivative.byTheory=cos(3);
Derivative.byDiference=(sin(3.01)-sin(3))/0.01; %前向差分近似
DefiniteIntegral,Derivative
fnplt(pp,'b-');hold on
fnplt(int_pp,'m:'),fnplt(der_pp,'r--');hold off
legend('y(x)','S(x)','dy/dx')
```

ppval(int_pp,[1,2])给出一个行向量，其中第一个元素是原函数在0～1的定积分，第2个元素是原函数在0～2的定积分，所以在1～2的定积分应是第2个元素减第1个元素的值

反应器停留时间分布的混合特性



在t = 0的时刻，在一容器入口处突然向流进容器的流体脉冲注入一定量的示踪剂，同时在容器出口处测量流出物料中示踪剂浓度随时间的变化，实验数据如下表:

t/s	0	20	40	60	80	100	120	140	160	180	200
$C(kmol/m^3)\times 10^3$	0	0	0	0	0.4	5.5	16.2	11.1	1.7	0.1	0

试计算流体在容器中的平均停留时间以及扩散准数。

数学模型:

平均停留时间

$$t_m = \frac{\int_0^\infty Ctdt}{\int_0^\infty Cdt}$$

方差:

$$\sigma^2 = \frac{\int_0^\infty Ct^2dt}{\int_0^\infty Cdt} - t_m^2$$

扩散特征数为:

$$\left(\frac{D_L}{uL}\right) = \frac{\sigma^2}{2t_m^2}$$

```
function Cha4demo7
t = [0:20:200];C = [0, 0, 0, 0, 0.4, 5.5, 16.2, 11.1, 1.7, 0.1, 0];plot(t,C,'o')
sp1= spline(t,C);
hold on
fnplt(sp1,'b-'); xlabel(‘Time (s)’)， ylabel('C (kmol/m^3)¡Á10^3')
hold off
% By spaps():
figure， plot(t,C,'o')
sp2= spaps(t,C,1);
hold on
fnplt(sp2,'b-'); xlabel(‘Time (s)’)， ylabel('C (kmol/m^3)¡Á10^3')
hold off
sp=input('Which function should be used to integrate?');
% Integration
t0 = 0;tf = t(end);
IC =  quadl(@Func,t0,tf,[],[],sp);
ICt = quadl(@Func1,t0,tf,[],[],sp);
ICt2 = quadl(@Func2,t0,tf,[],[],sp);
tm1 = ICt/IC
ss1 = ICt2/IC - tm1^2
DL2uL1 = ss1/tm1^2/2
% ------------------------------------------------------------
function y = Func(x,sp)     % f= C
y = fnval(sp,x);
% ------------------------------------------------------------
function y = Func1(x,sp)    % f= Ct
y = fnval(sp,x).*x;
% ------------------------------------------------------------
function y = Func2(x,sp)    % f= Ct^2
y = fnval(sp,x).*x.^2;
```

微分法进行动力学数据分析



反应物A在一等温间歇反应器中发生反应为 $A \rightarrow 产物$ 测量得到反应器中不同时间下反应物A的浓度C_A如下表所示。试根据表中数据确定其反应速率方程。

t/s	0	20	40	60	120	180	300
C_A(mol/L)	10	8	6	5	3	2	1

数学模型： $-r_A = kC_A^n$ → $\ln(-r_A) = n\ln C_A + \ln k$

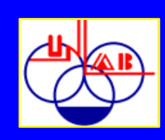
```
function Cha4demo8
% 动力学数据
t = [0 20  40  60  120  180  300];CA = [10  8  6  5  3  2  1];
% 用最小二乘样条拟合法计算微分dCA/dt
knots = 3;K = 3;      % 三次B样条
sp = spap2(knots,K,t,CA);
sp = spap2(newknt(sp),K,t,CA);
pp = fnder(sp);          % 计算B样条函数的导函数
dCAdt = fnval(pp,t);     % 计算t处的导函数值
% 绘制图形
ti = linspace(t(1),t(end),200);
CAi = fnval(sp,ti);
plot(t,CA,‘ro’,ti,CAi,‘b-’)， xlabel(‘t’)， ylabel('C_A')
figure
fnplt(pp);
% dCAdti = fnval(pp,ti)
% plot(ti,dCAdti,'-')
xlabel('t')
ylabel(‘dC/dt')
% 线性拟合
rA = dCAdt;
y = log(-rA);
x = log(CA);
p = polyfit(x,y,1);
k = exp(p(2))
n = p(1)
```

第五讲
常微分方程数值解

化工学院软件应用教科组

2006-10

本章知识要点



- 数值计算
  - 常微分方程初值问题
  - 常微分方程边值问题

- MATLAB
  - 微分方程求解常微分方程的相关函数
    - ode45 ode23
    - bvp4c

微分方程在化工模型中的应用



- 间歇反应器的计算
- 活塞流反应器的计算
- 全混流反应器的动态模拟
- 定态一维热传导问题
- 逆流壁冷式固定床反应器一维模型
- 固定床反应器的分散模型

Matlab常微分方程求解问题分类



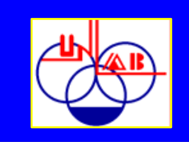
初值问题:

- 定解附加条件在自变量的一端
- 一般形式为: $\begin{cases} y' = f(x,y) \\ y(a) = y_0 \end{cases}$
- 初值问题的数值解法一般采用步进法，如Runge-Kutta法

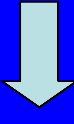
边值问题:

- 在自变量两端均给定附加条件
- 一般形式: $\begin{cases} y' = f(x,y) \\ y(a) = y_1, y(b) = y_2 \end{cases}$
- 边值问题可能有解、也可能无解，可能有唯一解、也可能有无数解
- 边值问题有3种基本解法
  - 迭加法
  - 打靶法
  - 松弛法

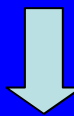
Matlab求解常微分方程初值问题方法



将待求解转化为标准形式，并"翻译"成Matlab可以理解的语言，即编写odefile文件



选择合适的解算指令求解问题



根据求解问题的要求，设置解算指令的调用格式

Matlab求解初值问题函数



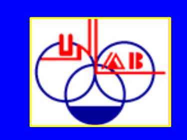
指令		含义	指令		含义
解算	ode23	普通2-3阶法解ODE		odefile	ODE文件格式
	ode45	普通4-5阶法解ODE	选项	odeset	创建、更改ODE选项的设置
	ode113	普通变阶法解ODE		odeget	读取ODE选项的设置
	ode23t	解适度刚性ODE	输出	odeplot	ODE的输出时间序列图
	ode15s	变阶法解刚性ODE		odephas2	ODE的二维相平面图
	ode23s	低阶法解刚性ODE		odephas3	ODE的三维相空间图
	ode23tb	低阶法解刚性ODE		odeprint	在Matlab指令窗显示结果

odefile



- 所谓的odefile实际上是一个Matlab函数文件，一般作为整个求解程序的一个子函数，表示ode求解问题
- Matlab提供了odefile的模板，采用type odefile命令显示其详细内容，然后将其复制到脚本编辑窗口，在合适的位置填入所需内容
- 一般而言，对于程序通用性要求不高的场合，只需将原有模型写成标准形式，然后“翻译”成Matlab语言即可

odefile的编写规定



1. ode文件的最简单格式必须有一个自变量t和函数y作为输入变量，一个y的导函数作为输出变量。其中自变量t不论在ode文件中是否使用都必须作为第一输入变量，y则必须作为第二输入变量，位置不能颠倒。
2. 可以向ode文件中传递参数，数目不受限制

odefile的编写



求解初值问题：

$$\begin{cases} y' = y - \dfrac{2x}{y} \\ y(0) = 1 \end{cases}$$

$(0 \le x \le 1)$

$$\begin{cases} y' = f(x, y) \\ y(a) = y_0 \end{cases}$$

ode输入函数

自变量在前，因变量在后

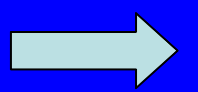
```
function f=fun(x,y)
f=y-2*x/y;
```

输出变量为因变量导数的表达式

初值问题：

$$\begin{cases} y' = y + y^2 \\ y(0) = 1 \end{cases}$$

$(0 \le x \le 1)$



```
function f=fun(x,y)
f=y + y^2;
```

常微分方程组odefile的编写

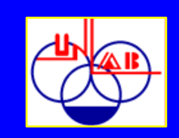


$$\begin{cases} y_1' = 0.04(1-y_1) - (1-y_2)y_1 + 0.0001(1-y_2)^2 \\ y_2' = -10^4 y_1' + 3000(1-y_2)^2 \\ y_1(0) = 0, y_2(0) = 1, 0 \le x \le 100 \end{cases}$$

常微分方程组与单个常微分方程求解方法相同，只需在编写odefile时将整个方程组作为一个向量输出。

```
function f=fun(x,y)
dy1dx = 0.04*(1-y(1))-(1-y(2)).*y(1)+0.0001*(1-y(2)).^2;
dy2dx = -1e4*dy1dx + 3000*(1-y(2)).^2;
f = [dy1dx; dy2dx];
```

高阶微分方程odefile的编写

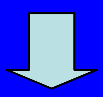


求解: $$y''+a(t)(y')^2+b(t)y=e^t\cos 2\pi t$$ y(0)=0，y'(0)=1，

$$a(t)=-e^{-t}+\cos 2\pi t e^{-2t}, b(t)=\cos(2\pi t)$$

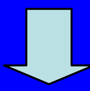
本例的难度:

方程系数非线性



可在odefile中定义

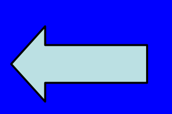
方程高阶，非标准形式



方程变形：令y1 = y； y2 = y'

则原方程等价于:

$$\begin{cases} y_1' = y_2 \\ y_2' = -ay_2^2 - by_1 + e^t\cos 2\pi t \end{cases}$$



```
function f=fun(t,y)
a=-exp(-t)+cos(2*pi*t)*exp(-2*t);
b=cos(2*pi*t);
f=[y(2)
   -a*y(2)^2-b*y(1)+exp(t)*b];
```

解算指令的使用方法



调用格式:

1. [T,Y]=ode45(@fun, TSPAN,Y0)
2. [T,Y]=ode45(@fun, TSPAN,Y0,options)
3. [T,Y]= ode45(@fun, TSPAN,Y0,options,P1,P2,…)
4. [T,Y,TE,YE,IE]= ode45(@fun, TSPAN,Y0,options,P1,P2,…)

说明:

❖ 输出变量T为返回时间列向量；解矩阵Y的每一行对应于T的一个元素，列数与求解变量数相等。

❖ @fun为函数句柄，为根据待求解的ODE方程所编写的ode文件（odefile）；

❖ TSPAN = [T0 TFINAL]是微分系统y' = F(t,y)的积分区间；Y0为初始条件

❖ options用于设置一些可选的参数值，缺省时，相对于第一种调用格式。options中可以设置的参数参见odeset

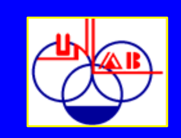
❖ P1，P2，…的作用是传递附加参数P1，P2，…到ode文件。当options缺省时，应在相应位置保留[]，以便正确传递参数。

常微分方程初值问题解算指令比较



解算指令	算法	精度
ode45	四五阶Runge-Kutta法	较高
ode23	二三阶Runge-Kutta法	低
ode113	可变阶Adams-Bashforth-Moulton法	
ode15s	基于数值差分的可变阶方法（BDFs，Gear）	低～中
ode23s	二阶改进的Rosenbrock法	低
ode23t	使用梯形规则	适中
ode23tb	TR-BDF2（隐式Runge-Kutta法）	低

ode解算指令的选择(1)



1.根据常微分方程要求的求解精度与速度要求

求解初值问题: $\begin{cases} y' = y - \dfrac{2x}{y} \\ y(0) = 1 \end{cases}$ $(0 \le x \le 1)$

比较ode45和ode23的求解精度和速度

ode45和ode23的比较



```
function Cha5demo1
%Comparison of results obtained from ode45 and ode23 solver
format long
y0=1;
tic,[x1,y1]=ode45(@fun,[0,1],y0);t_ode45=toc
tic,[x2,y2]=ode23(@fun,[0,1],y0);t_ode23=toc
plot(x1,y1,'b-',x2,y2,'m-.'),
xlabel('x'),ylabel('y'),
legend('ODE45','ODE23','location','Northwest')

disp('Comparative Results at x=1:');
fprintf('\nODE45\t\t\t y=%.8f\nODE23\t\t\t y=%.8f\nPrecisive Result ...
=%.8f\n',y1(end),y2(end),1.7320508)
%-----------------------------------------------------------------------
function f=fun(x,y)
f=y-2*x/y;
```

ode解算指令的选择(2)



2.根据常微分方程组是否为刚性方程

$$\begin{cases} y' = Ay + b(x) \\ y(x_0) = y_0 \end{cases}$$

如果系数矩阵A的特征值连乘积小于零，且绝对值最大和最小的特征值之比（刚性比）很大，则称此类方程为刚性方程

$$\begin{cases} y_1' = -0.01y_1 - 99.99y_2 \\ y_2' = -100y_2 \\ y_1(0) = 2, y_2(0) = 1 \end{cases}$$

刚性比：100/0.01 = 10000

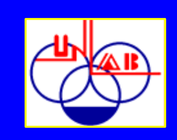
- 刚性方程在化学工程和自动控制领域的模型中比较常见。

ode解算指令的选择(2)



- 刚性方程的物理意义：常微分方程组所描述的物理化学变化过程中包含了多个子过程，其变化速度相差非常大的数量级，于是常微分方程组含有快变和慢变分量。
- 常微分方程组数值积分的稳定步长受模值最大的特征值控制，即受快变量分量约束，特征值大则允许步长小；而过程趋于稳定的时间又由模值最小的特征值控制，特征值小则积分到稳定的时间则长。
- Matlab提供了不同种类的刚性方程求解指令：ode15s ode23s ode23t ode23tb，可根据实际情况选用

刚性常微分方程组求解



```
function Cha5demo3
figure
ode23s(@fun,[0,100],[0;1])
hold on,
ode45(@fun,[0,100],[0;1])
%----------------------------------------------------------------
function f=fun(x,y)
dy1dx = 0.04*(1-y(1))-(1-y(2)).*y(1)+0.0001*(1-y(2)).^2;
dy2dx = -1e4*dy1dx + 3000*(1-y(2)).^2;
f = [dy1dx; dy2dx];
```

解算指令的输出控制



1. [T,Y]= ode45(@fun, TSPAN,Y0,options,P1,P2,…)
2. sol=ode45(@fun,TSPAN,Y0)将解输出指结构体sol中；SXINT = deval(SOL,XINT)用于计算解在XINT处的值，XINT必须位于区间[SOL.x(1) SOL.x(end)]内。
3. 在无输出变量时，将调用默认的odeplot输出解的图形。
4. 除了以odeplot形式输出外，还可以以odephas2，和odephas3的形式输出解向量的二维和三维相平面图。
5. 采用以下语句options=odeset(‘outputfcn’,‘odephas2’)可以将输出方法改变为平面输出
6. odeprint输出求解过程每一步的解

解算指令的options选项

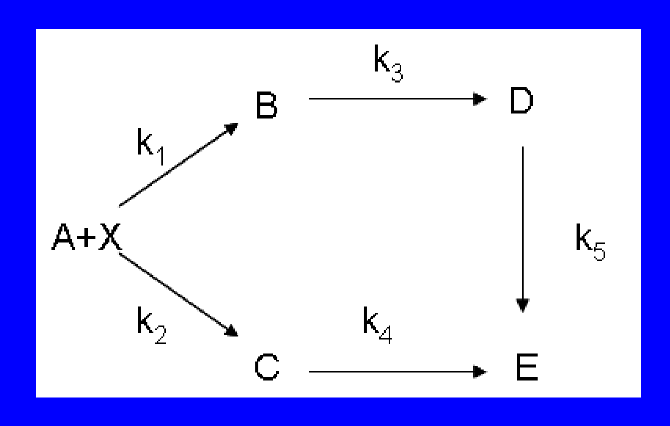


1. RelTol－相对误差，它应用于解向量的所有分量。在每一步积分过程中，第i个分量误差e(i)满足：e(i)<=max(RelTol*abs(y(i),AbsTol(i))。
2. AbsTol－绝对误差，若是实数，则应用于解向量的所有分量，若是向量，则它的每一个元素应用于对应位置解向量元素。
3. OutputFcn－可调用的输出函数名。每一步计算完后，这个函数将被调用输出结果，可以选择的值为：odeplot，odephas2，odephas3，odeprint。
4. OutputSel－输出序列选择。指定解向量的哪个分量被传递给OutputFcn。
5. MaxSetp－步长上界，缺省值为求解区间的1/10。
6. InitialStep－初始步长，缺省时自动设置。
7. 采用odeset改变原有选项的值

间歇反应器中串联－平行复杂反应系统



在间歇反应器中进行液相反应制备产物B，反应网络如图5-1所示。反应可在180~260℃的温度范围内进行，反应物X大量过剩， 而C, D和E为副产物。各反应均为一级动力学关系：r＝－kC，式中 $k = k_0 e^{\frac{-Ea}{RT}}$
已知参数：k_{01} = 5.78052 × 10^{10}，k_{02} = 3.92317 × 10^{12}，k_{03} = 1.64254 × 10^{4}，k_{04} = 6.264 × 10^{8}，Ea_1=124670，Ea_2=150386，Ea_3=77954，Ea_4=111528。初始浓度：C_A=1kmol/m^3，其余物质浓度为0。
已知是产物B收率最大的最优反应温度为224.6℃
试计算1)在最优反应温度下各组分浓度随时间的动态变化；2)最优反应时间；3)输出产物D对反应物浓度A的关系图。



间歇反应器中串联－平行复杂反应系统



数学模型

$$\frac{dC_A}{dt} = -(k_1 + k_2)C_A$$

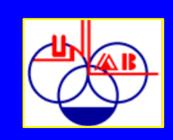
$$\frac{dC_B}{dt} = k_1 C_A - k_3 C_B$$

$$\frac{dC_C}{dt} = k_2 C_A - k_4 C_C$$

$$\frac{dC_D}{dt} = k_3 C_B - k_5 C_D$$

$$\frac{dC_E}{dt} = k_4 C_C + k_5 C_D$$

间歇反应器中串联－平行复杂反应系统



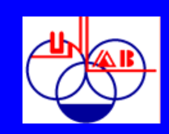
```
function Cha5demo4
T = 224.6 + 273.15; R = 8.31434; k0 = [5.78052E+10  3.92317E+12  1.64254E+4  6.264E+8];
Ea = [124670  150386  77954  111528];
% Initial concentration:C0(i), kmol/m^3
C0 = [1  0  0  0  0]; tspan = [0 1e4];
opt=odeset(‘reltol’,1e-4,’outputfcn’,’odephas2’,’outputsel’,[1;4])
[t,C] = ode45(@MassEquations, tspan, C0,opt,k0,Ea,R,T)
% plot
plot(t,C(:,1),'r-',t,C(:,2),'k:',t,C(:,3),'b-.',t,C(:,4),'k--');
xlabel('Time (s)');
ylabel('Concentration (kmol/m^3)');
legend('A','B','C','D')
CBmax = max(C(:,2));    % CBmax
yBmax = CBmax/C0(1)     % yBmax:
index = find(C(:,2)==CBmax);
t_opt = t(index)        % t_opt: the optimum batch time, s
% ------------------------------------------------------------
function dCdt = MassEquations(t,C,k0,Ea,R,T)
% Reaction rate constants, 1/s
k = k0.*exp(-Ea/(R*T)); k(5) = 2.16667E-04;
% Reaction rates, kmoles/m3 s
rA = -(k(1)+k(2))*C(1); rB = k(1)*C(1)-k(3)*C(2);rC = k(2)*C(1)-k(4)*C(3); rD = k(3)*C(2)-k(5)*C(4);
rE = k(4)*C(3)+k(5)*C(4);
% Mass balances
dCdt = [rA; rB; rC; rD; rE];
```

Matlab求解边值问题方法：bvp4c函数



1. 把待解的问题转化为标准边值问题 $\begin{cases} y' = f(x,y) \\ g(y(a),y(b)) = 0 \end{cases}$
2. 因为边值问题可以多解，所以需要为期望解指定一个初始猜测解。该猜测解网（Mesh）包括区间[a, b]内的一组网点（Mesh points）和网点上的解S(x)
3. 根据原微分方程构造残差函数 $r(x) = S'(x) - f(x, S(x))$
4. 利用原微分方程和边界条件，借助迭代不断产生新的S(x)，使残差不断减小，从而获得满足精度要求的解

Matlab求解边值问题的基本指令



solinit = bvpinit（x,v,parameters）

生成bvp4c调用指令所必须的“解猜测网”

sol = bvp4c（odefun，bcfun，solinit，options，p1，p2，…）

给出微分方程边值问题的近似解

sxint = deval（sol，xint）

计算微分方程积分区间内任何一点的解值

初始解生成函数：bvpinit()



solinit = bvpinit（x,v,parameters）

- x指定边界区间[a,b]上的初始网络，通常是等距排列的（1 × M）一维数组。注意：使x(1)=a，x(end)=b；格点要单调排列。
- v是对解的初始猜测
- solinit（可以取别的任意名）是“解猜测网（Mesh）”。它是一个结构体，带如下两个域:
  - ✧ solinit.x是表示初始网格有序节点的（1 × M）一维数组，并且solinit.x(1)一定是a，solinit.x(end)一定是b。M不宜取得太大，10数量级左右即可。
  - ✧ solinit.y是表示网点上微分方程解的猜测值的（N × M）二维数组。solinit.y(:,i)表示节点solinit.x(i)处的解的猜测值。

边值问题求解指令：bvp4c ()



sol = bvp4c（odefun，bcfun，solinit，options，p1，p2，…）

输入参数:

- odefun是计算导数的M函数文件。该函数的基本形式为：dydx = odefun(x,y,parameters,p1,p2，…)，在此，自变量x是标量，y，dydx是列向量。其基本形式与初值问题的odefile形式相同
- bcfun是计算边界条件下残数的M函数文件。其基本形式为：res = bcfun(ya,yb,parameters,p1,p2,……)，文件输入宗量ya,yb是边界条件列向量,分别代表y在a和b处的值。res是边界条件满足时的残数列向量。注意：例如odefun函数的输入宗量中包含若干“未知”和“已知”参数，那么不管在边界条件计算中是否用到，它们都应作为bcfun的输入宗量。
- 输入宗量options是用来改变bvp4c算法的控制参数的。在最基本用法中，它可以缺省，此时一般可以获得比较满意的边值问题解。如需更改可采用bvpset函数，使用方法同odeset函数。
- 输入宗量p1，p2等表示希望向被解微分方程传递的已知参数。如果无须向微分方程传递参数，它们可以缺省。

边值问题求解指令：bvp4c ()



输出参数:

输出变量sol是一个结构体

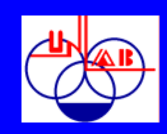
◆sol.x是指令bvp4c所采用的网格节点;

◆sol.y是y(x)在sol.x网点上的近似解值;

◆sol.yp是y'(x)在sol.x网点上的近似解值;

◆sol.parameters是微分方程所包含的未知参数的近似解值。当被解微分方程包含未知参数时，该域存在。

边值问题的求解



求解两点边值问题: $\begin{cases} y''-y=10 \\ y(0)=y(1)=0 \end{cases}$

令: $y_1 = y, y_2 = y_1'$

原方程组等价于以下标准形式的方程组:

$$\begin{cases} y_1' = y_2 \\ y_2' = y_1 + 10 \end{cases}$$

边界条件为:

$$y_1(0) = 0, y_1(1) = 0$$

```
function Cha5demo5
solinit=bvpinit(linspace(0,1,10),[1 0]);
sol=bvp4c(@ODEfun,@BCfun,solinit);
x=[0:0.05:0.5];
y=deval(sol,x);
yP=[0 -0.41286057 -0.729740656...
-0.95385538 -1.08743325 -1.13181116];
xP=[0:0.1:0.5];
plot(xP,yP,'o',x,y(1,:),'r-'),
legend('Analytical Solution','Numerical Solution')
% -----------------------------------------------------------------------
function dydx=ODEfun(x,y)
dydx=[y(2);y(1)+10];
% -----------------------------------------------------------------------
function bc=BCfun(ya,yb)
bc=[ya(1);yb(1)];
```

边值问题的求解



求解：

$$\begin{cases} y''-(1+x^2)y=1 \\ y(0)=1, y(1)=3 \end{cases}$$

令：

$$y_1=y, y_2=y_1'$$

原方程组等价于以下标准形式的方程组：

$$\begin{cases} y_1'=y_2 \\ y_2'=(1+x^2)y_1+1 \end{cases}$$

边界条件为：

$$\begin{cases} y_1(0)-1=0 \\ y_1(1)-3=0 \end{cases}$$

```
function Cha5demo6
solinit = bvpinit(linspace(0,1,10),[0 1]);
sol = bvp4c(@ODEfun,@BCfun,solinit);
x = [0:0.1:1];
y = deval(sol,x);
yP=[1 1.0743 1.1695 1.2869 1.4284...
1.5965 1.7947 2.0274 2.3004 2.6214 3];
xP=[0:0.1:1.0];
plot(xP,yP,'o',x,y(1,:),'r-')
legend('Analytical Solution','Numerical Solution','location','Northwest')
legend boxoff
% -------------------------------------------------------------
function dydx = ODEfun(x,y)
dydx = [y(2); (1+x^2)*y(1)+1];
% -------------------------------------------------------------
function bc = BCfun(ya,yb)
bc = [ya(1)-1;yb(1)-3];
```

边值问题的求解



求解:

$$z'' + c \cdot |z| = 0 \quad c = 1$$

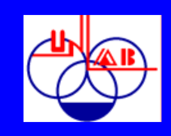
$$z(0) = 0, z(4) = -2$$

令: $z_1 = z, z_2 = z_1'$

```
function Cha5demo7
c=1;
solinit = bvpinit(linspace(0,4,10),[1;1]);
sol = bvp4c(@ODEfun,@BCfun,solinit,[],c);
x = [0:0.1:4];
y = deval(sol,x);
plot(x,y(1,:),'b-',sol.x,sol.y(1,:),'ro')
legend('解曲线','初始网格点解')
% ------------------------------------------------------------------
function dydx = ODEfun(x,y,c)
dydx = [y(2); -c*abs(y(1))];
% ------------------------------------------------------------------
function bc = BCfun(ya,yb)
bc = [ya(1);yb(1)+2];
```

此程序有什么错误?

边值问题的求解



求解: $z'' + (\lambda - 2q\cos 2x)z = 0$

$q = 15$

边界条件:

$$z(0) = 1, z'(0) = 0, z'(\pi) = 0$$

```
function Cha5demo8
lmb=15;
solinit = bvpinit(linspace(0,pi,10),[1;1],lmb);
opts=bvpset('Stats','on');
sol = bvp4c(@ODEfun,@BCfun,solinit,opts);
lambda=sol.parameters
x = [0:pi/60:pi];
y = deval(sol,x);
plot(x,y(1,:),'b-',sol.x,sol.y(1,:),'ro')
legend('解曲线','初始网格点解')
% ---------------------------------------------------------------------------------
function dydx = ODEfun(x,y,lmb)
q=5;
dydx = [y(2); -(lmb-2*q*cos(2*x))*y(1)];
% ---------------------------------------------------------------------------------
function bc = BCfun(ya,yb,lmb)
bc = [ya(1)-1;ya(2);yb(2)];
```

本例中，微分方程与参数λ的数值有关。一般而言，对于任意的λ值，该问题无解，但对于特殊的λ值（特征值），它存在一个解，这也称为微分方程的特征值问题。对于此问题，可在bvpinit中提供参数的猜测值，然后重复求解BVP得到所需的参数，返回参数为sol.parameters

边值问题的求解



求解：

$$\begin{cases} u' = 0.5u(w-u)/v \\ v' = -0.5(w-u) \\ w' = (0.9 - 1000(w-y) - 0.5w(w-u))/z \\ z' = 0.5(w-u) \\ y' = -100(y-w) \end{cases}$$

边界条件：

$$u(0) = v(0) = w(0) = 1, \quad z(0) = -10, w(1) = y(1)$$

初始猜测解为：

$$u(x) = 1, v(x) = 1, w(x) = -4.5x^2 + 8.91x + 1, \quad z(x) = -10, y(x) = -4.5x^2 + 9x + 0.91$$

```
function Cha5demo9
solinit = bvpinit(linspace(0,1,5),@ex1init);
options = bvpset('Stats','on','RelTol',1e-5);
sol = bvp4c(@ODEfun,@BCfun,solinit,options);
x = sol.x;
y = sol.y;
........
% ------------------------------------------------------------------
function dydx = ODEfun(x,y)
dydx =  [ 0.5*y(1)*(y(3) - y(1))/y(2)
         -0.5*(y(3) - y(1))
         (0.9 - 1000*(y(3) - y(5)) - 0.5*y(3)*(y(3) - y(1)))/y(4)
          0.5*(y(3) - y(1))
          100*(y(3) - y(5)) ];
%------------------------------------------------------------------
function bc = BCfun(ya,yb)
bc = [ ya(1) - 1
        ya(2) - 1
        ya(3) - 1
        ya(4) + 10
        yb(3) - yb(5)];
%------------------------------------------------------------------
 function v = ex1init(x)
v = [1;   1;    -4.5*x^2+8.91*x+1;   -10;    -4.5*x^2+9*x+0.91 ];
```

边值问题的求解



求解： $f''' + ff'' + \beta(1-(f')^2) = 0$

$\beta = 0.5$

边界条件： $f(0)=0,\ f'(0)=0,$
$f'(\eta)\to 1\ \eta\to\infty$

如果取1，计算结果如何

```
function Cha5demo11
infinity = 6;
solinit = bvpinit(linspace(0,infinity,5),[0 0 1]);
options = bvpset('stats','on');
sol = bvp4c(@ex5ode,@ex5bc,solinit,options);
eta = sol.x;
f = sol.y;
fprintf('\n');
fprintf('Cebeci & Keller report f''(0) = 0.92768.\n')
fprintf('Value computed here is f''(0) = %7.5f.\n',f(3,1))
clf reset
plot(eta,f(2,:));
axis([0 infinity 0 1.4]);
title('Falkner-Skan equation, positive wall shear, \beta = 0.5.')
xlabel('\eta'),ylabel('df/d\eta'),shg
% --------------------------------------------------------------------------
function dfdeta = ex5ode(eta,f)
beta = 0.5;
dfdeta = [ f(2);   f(3);   -f(1)*f(3) - beta*(1 - f(2)^2) ];
% --------------------------------------------------------------------------
function res = ex5bc(f0,finf)
res = [f0(1);   f0(2);   finf(2) - 1];
```

常微分方程边值问题的应用



已知在球形氧化铝催化剂进行环己烷脱氢反应，催化剂直径为5mm，操作温度为700K，在此温度下，$k = 4s^{-1}$，$D=0.05cm^2/s$。计算催化剂颗粒内环己烷的浓度分布及催化剂的有效因子。

数学模型:

质量传递方程

$$\frac{d^2C}{dr^2}+\frac{2}{r}\frac{dC}{dr}=\phi^2\frac{f(C)}{C_S}$$

$0<r<1$

Thiele模数

$$\phi=r_P\sqrt{\frac{k}{D}}$$

边界条件:

$$\frac{dC}{dr}(0)=0, C(1)=1$$

等温有效因子为:

$$\eta=\frac{\int_0^1 f(C)r^2dr}{\int_0^1 f(C_S)r^2dr}$$

```
function Cha5demo12
 global fai Cs
Cs = 1; rp = 5e-3/2; k = 4; D = 0.05e-4; fai = rp*sqrt(k/D)  % fai: Thiele模数
a = 0;b = 1;
solinit = bvpinit(linspace(a,b,100),[0 0]);
sol = bvp4c(@ODEs,@BCfun,solinit);
plot(sol.x,sol.y(1,:),'b-')
xlabel(‘Dimensionless Radius’)，ylabel('Dimensionless Concentration')
I1 = quadl(@IntFunc1,a,b,[],[],sol.x,sol.y(1,:))
I2 = quadl(@IntFunc2,a,b)
eta = I1/I2
fprintf('\t催化剂的有效因子为: η = %.4f\n',eta)
% ------------------------------------------------------------
function dydr = ODEs(r,y)
global fai Cs
dy1dr = y(2);
if r == 0
    dy2dr = 0;
else
    dy2dr = fai^2*y(1)/Cs - 2/r*y(2);
end
dydr = [dy1dr; dy2dr];
% ------------------------------------------------------------
function bc = BCfun(ya,yb)
bc = [yb(1)-1; ya(2)];
% ------------------------------------------------------------
function f = IntFunc1(r,ri,yi)
f = spline(ri,yi,r) .* r.^2;
% ------------------------------------------------------------
function f = IntFunc2(r)
global Cs
f = Cs * r.^2;
```

化工数学模型中的微分方程



1. 集中参数模型和分布参数模型

   集中参数模型是指因变量不随空间坐标变化，模型中自变量仅为时间；分布参数模型则指因变量不仅与时间有关，而且与空间有关。集中参数模型为常微分方程，分布参数模型为偏微分方程。

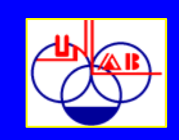
2. 化工模型中最常见的常微分方程边值问题模型为涉及传热和扩散的问题
3. 在建立微分方程模型时，请注意初始和边界条件，这样才是一个完整的定解问题
4. 化工模型中的微分方程种类很多，但其求解过程往往并不复杂。

第六讲
偏微分方程数值解

化工学院软件应用教科组

2006-10

本章知识要点



- MATLAB求解偏微分方程函数－pdepe的使用方法
- Matlab的PDE Toolbox图形用户界面（GUI）求解偏微分方程
- PDE Toolbox命令行编程求解偏微分方程

偏微分方程在化工模型中的应用



- 传热、传质过程主要是抛物型方程
- 椭圆型方程（稳态方程）
- 流体流动方程多数为双曲型方程

偏微分方程分类



线性偏微分方程:

方程经过有理化并消去分式后，若方程中没有未知函数及其偏导数的乘积或幂等非线性项，那就称之为线性偏微分方程（组）。如果有则称为非线性偏微分方程

拟线性偏微分方程:

如果仅对未知函数的所有最高阶导数来说是线性的，则称之为拟线性偏微分方程（组）。

齐次与非齐次方程:

在线性方程中，不含有未知函数及其偏导数的项称为自由项，自由项为零的方程称为齐次方程，否则称为非齐次方程。

二阶偏微分方程的分类



$$A\frac{\partial^2 u}{\partial x^2}+B\frac{\partial^2 u}{\partial xy}+C\frac{\partial^2 u}{\partial y^2}+D\frac{\partial u}{\partial x}+E\frac{\partial u}{\partial y}+Fu=f(x,y,u,\frac{\partial u}{\partial x},\frac{\partial u}{\partial y})$$

当式中A，B，C为常数时，即为拟线性偏微分方程，可以分为三类:

当$B^2-4AC<0$时，方程称为椭圆型（elliptic）偏微分方程;

当$B^2-4AC=0$时，方程称为抛物型（parabolic）偏微分方程;

当$B^2-4AC>0$时，方程称为双曲型（hyperbolic）偏微分方程。

三个最基本最典型的偏微分方程



双曲型方程（波动）方程

$$\frac{\partial^2 u}{\partial t^2} = a^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)$$

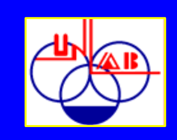
椭圆形方程（拉普拉斯方程）

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = 0$$

抛物型方程（热传导方程）

$$\frac{\partial u}{\partial t} = a^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)$$

偏微分方程的定解条件



初始条件：$U(x,t)\big|_{t=0} = \phi(x)$

对应初始条件的定解问题称为柯西（Cauchy）问题

边界条件：

第一种边界条件：$u(x,t)\big|_{t=0} = \phi(x)$　　对应的定解问题称为Dirichlet问题

第二种边界条件：$\dfrac{\partial u}{\partial n}\Big|_{x\in\Gamma} = f(x,t)$　　对应的定解问题称为Neumann问题

第三种边界条件：$(\dfrac{\partial u}{\partial n} + \sigma u)\Big|_{x\in\Gamma} = f(x,t)$　　对应的定解问题称为Robin问题

既有初始条件又有边界条件的定解问题称为混合问题

偏微分方程的数值解法



有限差分法：

微分方程的导数用有限差分近似代替，最后得到代数方程组

求解偏微分方程最基本的方法

正交配置法:

采用正交多项式为试函数，构造其系数使其近似于微分方程近似解

求解边值问题的有效方法

线上法：

将一个自变量当成连续变量，而对其自变量用有限差分法或正交配置法进行离散，从而把偏微分方程转变为常微分方程组

有限元法：

变分原理和分片插值的结合体，将求解区域分成有限个单元，然后采用分片插值函数逼近解，从而得到代数方程组。

线上法（MOL）求解偏微分方程示例



一维热传导方程 $\frac{\partial T}{\partial t}=2\frac{\partial^2 T}{\partial x^2}, x\in[0,10], t\geq 0$

初始条件: $u(x,0)=0$　　边界条件: $u(0,t)=100, u(10,t)=0$

- 在x方向，将求解区域5等份，则T的值可用中间四个节点处的值表示
- T对x的2阶导数以差分代替，即 $\frac{\partial^2 T}{\partial x^2}=\frac{T_{m-1}-2T_m+T_{m+1}}{(\Delta x)^2}$
- 原方程求解转化为以下常微分方程组的初值问题

$$\frac{\partial T_1}{\partial t}=2\frac{T_0-2T_1+T_2}{(\Delta x)^2}$$

$$\frac{\partial T_2}{\partial t}=2\frac{T_1-2T_2+T_3}{(\Delta x)^2}$$

$$\frac{\partial T_3}{\partial t}=2\frac{T_2-2T_3+T_4}{(\Delta x)^2}$$

$$\frac{\partial T_4}{\partial t}=2\frac{T_3-2T_4+T_5}{(\Delta x)^2}$$

Matlab求解偏微分方程方法



pdepe函数

- 求解一维动态模型，即一维抛物型和椭圆型偏微分方程的初值－边值问题
- 采用线上法求解，利用正交配置进行空间离散

pdetool图形用户界面

- 求解二维线性和非线性偏微分方程的PDE工具箱
- 有限元法
- 可用于求解具有标准形式的偏微分方程

pdepe求解偏微分方程的形式



偏微分方程组

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial t}=x^{-m}\frac{\partial}{\partial x}\left(x^{m}f\left(x,t,u,\frac{\partial u}{\partial x}\right)\right)+s\left(x,t,u,\frac{\partial u}{\partial x}\right)$$

为对角阵，该对角阵的元素是零(椭圆型)或者正数(抛物型方程)必须至少有一个方程是抛物型方程才可使用pdepe求解。

通量项

源项

m = 0，1，2分别代表平板，圆柱和球形

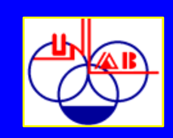
初始条件

$$u(x,t_0)=u_0(x)$$

边界条件

$$p(x,t,u)+q(x,t)f\left(x,t,u,\frac{\partial u}{\partial x}\right)=0$$

pdepe函数的调用



调用格式:

sol=pdepe(m, pdepe, icfun, bcfun, xmesh, tspan)

sol=pdepe(m, pdepe, icfun, bcfun, xmesh, tspan, options)

sol=pdepe(m, pdepe, icfun, bcfun, xmesh, tspan, options, p1, p2, …)

输出参数:

输出变量sol是三维数组

- sol(:,:,i)=ui是解向量的第i个分量
- sol(j,k,i)=u(j,k)是解向量ui在(x,t)=(tspan(j),xmesh(k))处的数值解
- sol(j,:,i)是解向量第i个分向量ui在时间tspan(j)和网格点xmesh(:)处的数值解

pdepe函数的输入参数



输入参数:

1. m表示定解问题的对称性，m=0, 1, 2分别代表平板、圆柱和球形。
2. pdefun 是描述PDE问题的函数，输出变量为标准形式中的c, f和s，其格式为[c, f, s]=pdefunc(x,t,u,dudx)。输入变量x和t为标量，u和dudx是向量。u和dudx分别是问题的解u(x,t)和它对x的偏导数的近似。c, f, s是列向量。
3. icfun描述定解问题初始条件的函数，其格式为u=icfun(x)。icfun计算和返回解的初始值。
4. bcfun是描述定解问题边界条件的函数，格式为[pl, ql, pr, qr]=bcfun(xl, ul, xr, ur, t)。其中ul是在左边界xl=a处的近似解，ur是在右边界xr=b处的近似解。pl和ql是p和q在xl处的列向量值，同样pr和qr是p和q在xr处的列向量值
5. xmesh是空间变量x的网点向量，pdepe不会自动选择。xmesh=[x0, x1, …, xn]，满足x0<x1<…<xn，且xmesh的长度必须大于3，xmesh(1)=a和xmesh(end)=b。pdepe的求解效率与xmesh的选择好坏关系很大。通常在梯度较大处应加密网格。
6. tspan是时间变量t的网点向量，是用户在该处想得到数值解的时间向量。tspan=[t0, t1, …, tn]，满足t0<t1<…<tf，且tspan的长度必须大于3，tspan(1)=t0和tspan(end)=tf。求解效率和tspan的疏密关系不大。

pdeval函数的使用



- 用于计算在区间[a,b]内的函数值和对x的偏导数值
- 调用格式：[xout,duoutdx]=pdeval(m, xmesh, ui, xout)
- ui=sol(j,:,i) 是问题解的第i个分量在时间tf和网点xmesh处的值。
- xout是[x0,xn]=[a,b]内的数构成的向量
- uout和duoutdx分别是 $u_i(x,t)$ 和 $\partial u_i(x,t)/\partial x$ 在xout处的函数值

一维热传导方程求解



一维热传导方程

$$\frac{\partial u}{\partial t} = \pi^{-2} \frac{\partial^2 u}{\partial x^2}, x \in [0,1], t \geq 0$$

初始条件: $u(x,0) = \sin(\pi x)$

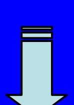
边界条件: $u(0,t) = 0, \pi e^{-t} + \frac{\partial u}{\partial x}(1,t) = 0$

一维热传导方程求解

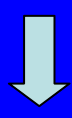


原始方程

$$\frac{\partial u}{\partial t} = \pi^{-2}\frac{\partial^2 u}{\partial x^2}, x \in [0,1], t \geq 0$$

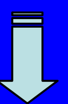


$$\pi^2\frac{\partial u}{\partial t} = x^{-0}\frac{\partial}{\partial x}(x^0\frac{\partial u}{\partial x}) + 0$$



初始条件

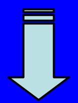
$$u(x,0) = \sin(\pi x)$$



$$u(x,0) = \sin(\pi x)$$

边界条件

$$u(0,t) = 0, \pi e^{-t} + \frac{\partial u}{\partial x}(1,t) = 0$$



$$u(0,t) + 0\cdot\frac{\partial u}{\partial x}(0,t) = 0$$

$$\pi e^{-t} + 1\cdot\frac{\partial u}{\partial x}(1,t) = 0$$

$m = 0$

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right) = \pi^2$$

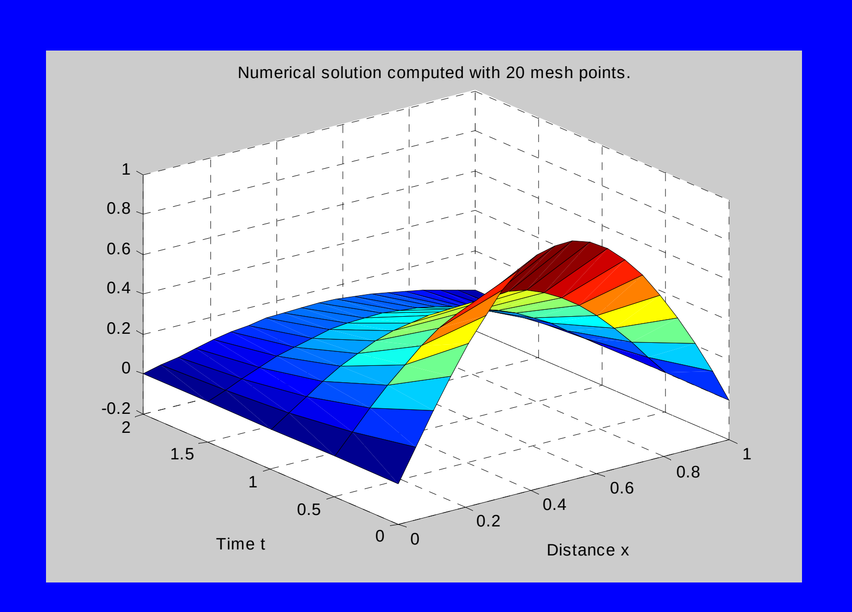
$$f\left(x,t,u,\frac{\partial u}{\partial x}\right) = \frac{\partial u}{\partial x}$$

$$s\left(x,t,u,\frac{\partial u}{\partial x}\right) = 0$$

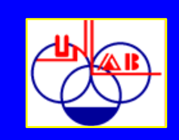
一维热传导方程求解



```
function Cha6demo1
m = 0;
x = linspace(0,1,20);
t = linspace(0,2,5);
sol = pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);
% Extract the first solution component as u.
u = sol(:,:,1);
% A surface plot is often a good way to study a solution.
surf(x,t,u)
title('Numerical solution computed with 20 mesh points.')
xlabel('Distance x')
ylabel('Time t')
% A solution profile can also be illuminating.
figure
plot(x,u(end,:)),title('Solution at t = 2')
xlabel('Distance x'),ylabel('u(x,2)')
% -----------------------------------------------------------
function [c,f,s] = pdex1pde(x,t,u,DuDx)
c = pi^2;f = DuDx;s = 0;
% -----------------------------------------------------------
function u0 = pdex1ic(x)
u0 = sin(pi*x);
% -----------------------------------------------------------
function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t)
pl = ul;ql = 0;pr = pi * exp(-t);qr = 1;
```



偏微分方程求解



定解问题

$$\frac{\partial u_1}{\partial t} = 0.024\frac{\partial^2 u_1}{\partial x^2} - F(u_1 - u_2)$$

$$\frac{\partial u_2}{\partial t} = 0.170\frac{\partial^2 u_2}{\partial x^2} + F(u_1 - u_2)$$

$$F(u) = \exp(5.73u) - \exp(-11.46u),$$
$$x \in [0,1], t \ge 0$$

$$u_1(x,0) = 1, u_2(x,0) = 0$$

$$\frac{\partial u_1}{\partial x}(0,t) = 0, u_2(0,t) = 0$$

$$\frac{\partial u_2}{\partial x}(1,t) = 1, u_1(1,t) = 0$$

标准形式

$$\begin{pmatrix}1\\1\end{pmatrix} \cdot \frac{\partial}{\partial t}\begin{pmatrix}u_1\\u_2\end{pmatrix} = \frac{\partial}{\partial x}\begin{pmatrix}0.024\partial u_1/\partial x\\0.170\partial u_2/\partial x\end{pmatrix} + \begin{pmatrix}-F(u_1-u_2)\\F(u_1-u_2)\end{pmatrix}$$

$$u_1(x,0) = 1, u_2(x,0) = 0$$

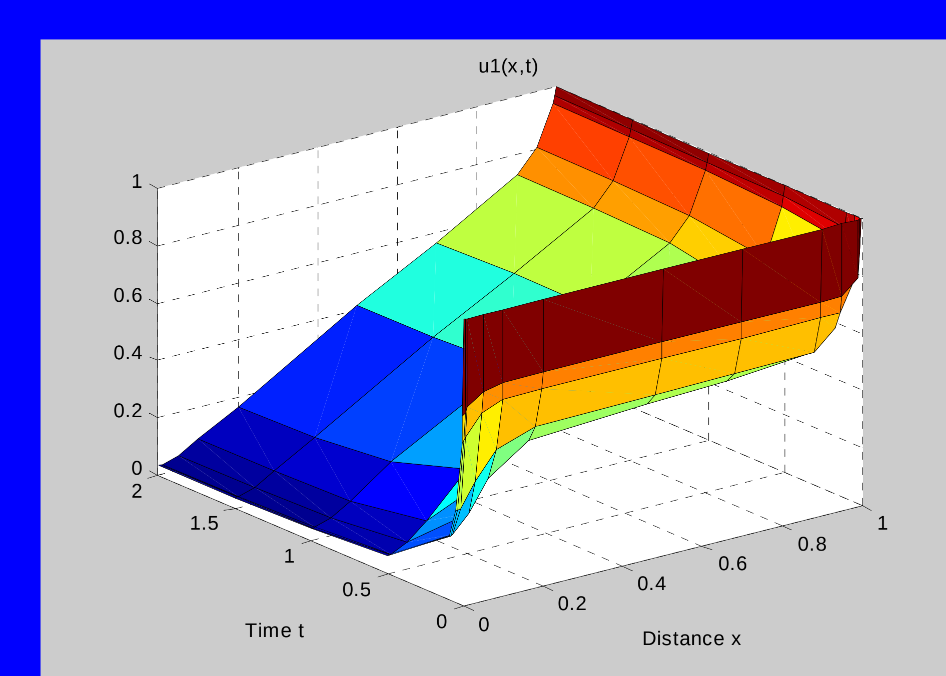
$$\begin{pmatrix}0\\u_2\end{pmatrix} + \begin{pmatrix}1\\0\end{pmatrix} \cdot \begin{pmatrix}0.024\partial u_1/\partial x\\0.170\partial u_2/\partial x\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}$$

$$\begin{pmatrix}u_1-1\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} \cdot \begin{pmatrix}0.024\partial u_1/\partial x\\0.170\partial u_2/\partial x\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}$$

偏微分方程求解



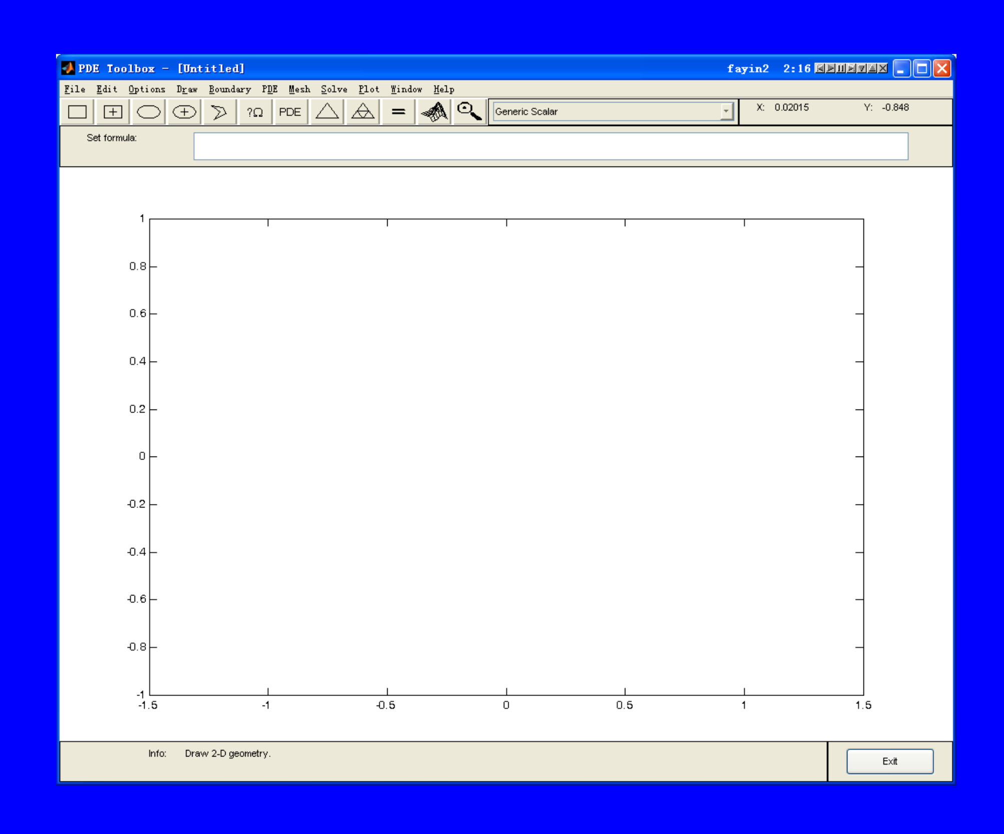
```
function Cha6demo2
m = 0;
x = [0 0.005 0.01 0.05 0.1 0.2 0.5 0.7 0.9 0.95 0.99 0.995 1];
t = [0 0.005 0.01 0.05 0.1 0.5 1 1.5 2];
sol = pdepe(m,@pdex4pde,@pdex4ic,@pdex4bc,x,t);
u1 = sol(:,:,1);u2 = sol(:,:,2);
figure
surf(x,t,u1),title('u1(x,t)'),xlabel('Distance x'),ylabel('Time t')
figure
surf(x,t,u2)
title('u2(x,t)'),xlabel('Distance x'),ylabel('Time t')
% -----------------------------------------------------------
function [c,f,s] = pdex4pde(x,t,u,DuDx)
c = [1; 1];
f = [0.024; 0.17] .* DuDx;
y = u(1) - u(2);
F = exp(5.73*y)-exp(-11.47*y);
s = [-F; F];
% -----------------------------------------------------------
function u0 = pdex4ic(x);
u0 = [1; 0];
% -----------------------------------------------------------
function [pl,ql,pr,qr] = pdex4bc(xl,ul,xr,ur,t)
pl = [0; ul(2)];
ql = [1; 0];
pr = [ur(1)-1; 0];
qr = [0; 1];
```



PDE Toolbox的图形用户界面



◆ 在命令窗口键入pdetool，则进入如右图所示的图形用户界面（GUI）



GUI能求解的各类偏微分方程形式



椭圆型方程: $-\nabla\cdot(c\nabla u)+au = f$

抛物型方程: $d\frac{\partial u}{\partial t}-\nabla\cdot(c\nabla u)+au = f$

双曲型方程: $d\frac{\partial^2 u}{\partial t^2}-\nabla\cdot(c\nabla u)+au = f$

特征值方程: $-\nabla\cdot(c\nabla u)+au = \lambda du$

非线性椭圆型方程: $-\nabla\cdot(c(u)\nabla u)+a(u)u = f(u)$

偏微分方程组:

$$
\begin{aligned}
-\nabla\cdot(c_{11}\nabla u_1)-\nabla\cdot(c_{12}\nabla u_2)+a_{11}u_1+a_{12}u_2 &= f_1 \\
-\nabla\cdot(c_{21}\nabla u_1)-\nabla\cdot(c_{22}\nabla u_2)+a_{21}u_1+a_{22}u_2 &= f_2
\end{aligned}
$$

GUI求解偏微分方程的定解条件



边界条件：

Dirichlet: $hu = r$ on the boundary $\partial\Omega$.

Generalized Neumann: $\vec{n}\cdot(c\nabla u) + qu = g$ on $\partial\Omega$.

- 在非线性条件下h,r,q,g可以是u的函数；对于抛物型和双曲型方程而言，可以是时间t的函数

对于二维体系：

Dirichlet:

$$
\begin{aligned}
h_{11}u_1 + h_{12}u_2 &= r_1 \\
h_{21}u_1 + h_{22}u_2 &= r_2
\end{aligned}
$$

Neumann:

$$
\begin{aligned}
\vec{n}\cdot(c_{11}\nabla u_1) + \vec{n}\cdot(c_{12}\nabla u_2) + q_{11}u_1 + q_{12}u_2 &= g_1 \\
\vec{n}\cdot(c_{21}\nabla u_1) + \vec{n}\cdot(c_{22}\nabla u_2) + q_{21}u_1 + q_{22}u_2 &= g_2
\end{aligned}
$$

混合边界条件：

$$
\begin{aligned}
h_{11}u_1 + h_{12}u_2 &= r_1 \\
\vec{n}\cdot(c_{11}\nabla u_1) + \vec{n}\cdot(c_{12}\nabla u_2) + q_{11}u_1 + q_{12}u_2 &= g_1 + h_{11}\mu \\
\vec{n}\cdot(c_{21}\nabla u_1) + \vec{n}\cdot(c_{22}\nabla u_2) + q_{21}u_1 + q_{22}u_2 &= g_2 + h_{12}\mu
\end{aligned}
$$

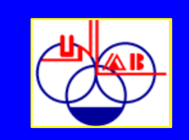
GUI求解偏微分方程的步骤



◆采用GUI求解偏微分方程，其步骤如下:

1. 画求解区域
2. 设置边界条件
3. 设置方程
4. 网格剖分
5. 解方程
6. 图形结果
7. 输出网格
8. 输出解的数值

GUI界面的菜单

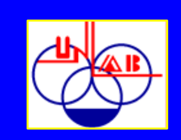


1. File
2. Edit
3. Options
4. Draw
5. Boundary
6. PDE
7. Mesh
8. Solve
9. Plot
10. Window
11. Help

基本操作（1–3）

求解过程（4–9）

GUI的工具栏



$\partial\Omega$	Enters the boundary mode.
PDE	Opens the PDE Specification dialog box.
	Initializes the triangular mesh.
	Refines the triangular mesh.
=	Solves the PDE.
	3-D solution opens the Plot Selection dialog box.

GUI求解椭圆方程示例



PDE Toolbox - [Untitled]

File Edit Options Draw Boundary PDE Mesh Solve Plot Window Help

Generic Scalar

X: -0.6115 Y: 1.102

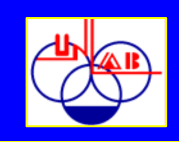
Set formula: C1

Color: u Vector field: -grad(u)

Info: Push to initialize mesh.

Exit

GUI求解抛物型方程示例



热传导方程：金属板导热问题

一个带有矩形孔的金属板，板的左边保持在100℃，右边热量从板向环境空气定常流动，其它边及内孔边界保持绝热。初始时板的温度为0℃。

金属板的外边界坐标为(-0.5,-0.8)，(0.5, -0.8), (0.5,0.8)，(-0.5,0.8)，内边界坐标为(-0.05, -0.4), (-0.05, -0.4), (-0.05, -0.4), (-0.05, -0.4)

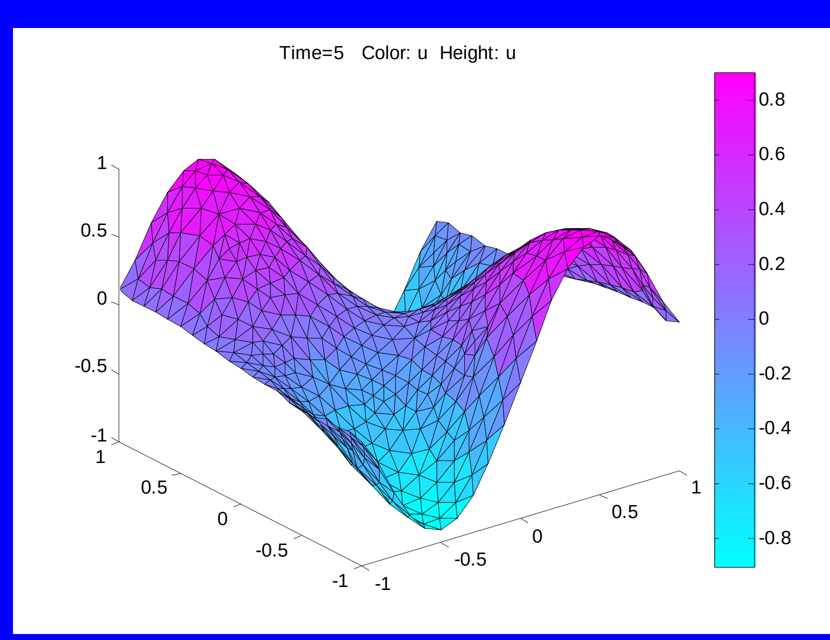
$$\begin{cases} d\dfrac{\partial u}{\partial t}-\Delta u=0, \\ u=100,\text{左边界} \\ \dfrac{\partial u}{\partial n}=-1,\text{右边界} \\ \dfrac{\partial u}{\partial n}=0,\ \text{其它边界} \\ u\big|_{t=t_0}=0 \end{cases}$$

GUI求解双曲型方程示例



二维波动方程的定解问题:

$$
\begin{cases}
\dfrac{\partial^2 u}{\partial t^2} - \Delta u = 0, (x, y) \in \Omega = \{(x, y) | -1 < x < 1, -1 < y < 1\} \\
u|_{x=\pm 1} = 0, \\
\dfrac{\partial u}{\partial n}\Big|_{y=\pm 1} = 0, \\
u = \arctan\left(\cos\left(\dfrac{\pi}{2}\right)\right), t = 0 \\
\dfrac{\partial u}{\partial t} = 3\sin(\pi x) e^{\sin(\frac{\pi}{2} y)}, t = 0
\end{cases}
$$

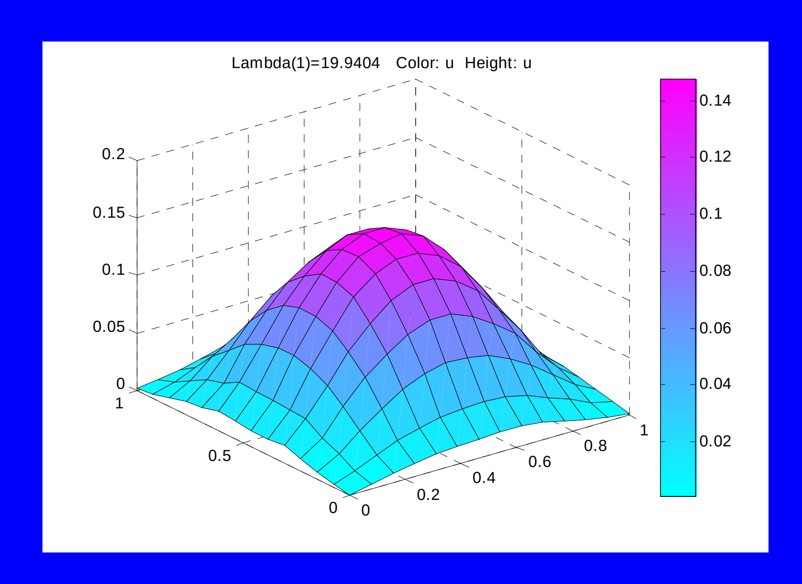


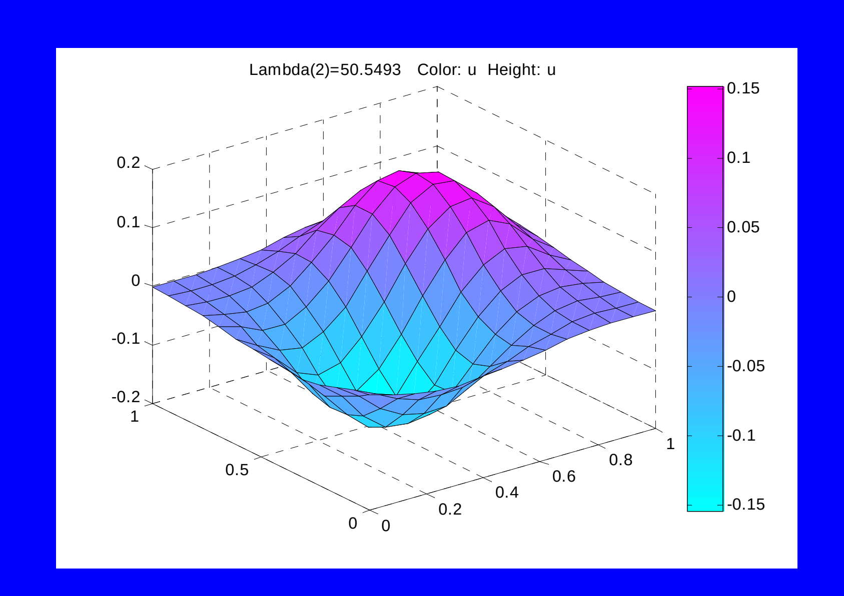
GUI求解特征值方程示例



求解方域上的特征值问题:

$$\begin{cases} -\Delta u = \lambda u, \Omega = \{(x,y) | 0 < x, y < 1\} \\ u|_{\partial\Omega} = 0 \end{cases}$$





有限元法的基本思想



椭圆方程：$-\nabla\cdot(c\nabla u)+au = f \text{ in } \Omega$

边界条件：Dirichlet: $hu = r$ on the boundary $\partial\Omega$.

Generalized Neumann: $\vec{n}\cdot(c\nabla u) + qu = g$ on $\partial\Omega$.

- 求解区域划分为多个子区域（有限元），在每个单元里以某种简单的函数来代替未知函数u
- 如果u是方程的解以任意的试函数v同乘方程两端，并积分：

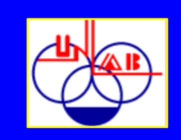
$$\int_\Omega -(\nabla\cdot(c\nabla u))v + auv \; dx = \int_\Omega fv\,dx$$

- 上式分步积分，并代入边界条件，则原求解问题变换为如下问题：求解u满足下式：

$$\left(\int_\Omega (c\nabla u)\cdot\nabla v + auv - fv \; dx - \int_{\partial\Omega}(-qu+g)v\,ds\right)=0 \qquad \forall v$$

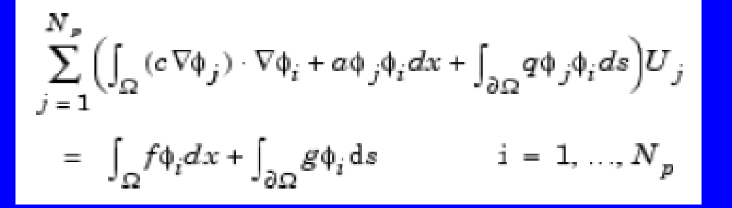
变分方程

有限元法的基本思想

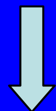


令：

$$u(x) = \sum_{j=1}^{N_p} U_j \phi_j(x$$



$$\sum_{j=1}^{N_p}\left(\int_\Omega (c\nabla\phi_j)\cdot\nabla\phi_i + a\phi_j\phi_i dx + \int_{\partial\Omega} q\phi_j\phi_i ds\right)U_j = \int_\Omega f\phi_i dx + \int_{\partial\Omega} g\phi_i ds \qquad i = 1, \ldots, N_p$$



$$K_{i,j} = \int_\Omega (c\nabla\phi_j)\cdot\nabla\phi_i dx$$ 刚度矩阵

$$M_{i,j} = \int_\Omega a\phi_j\phi_i dx$$ 质量矩阵

$$Q_{i,j} = \int_{\partial\Omega} q\phi_j\phi_i ds$$

$$F_i = \int_\Omega f\phi_i dx$$

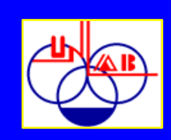
$$G_i = \int_{\partial\Omega} g\phi_i ds$$

$$(K + M + Q)U = F + G$$

抛物方程 $$M\frac{dU}{dt} + KU = F$$

双曲方程 $$M\frac{d^2U}{dt^2} + KU = F$$

PDE Toolbox命令行求解的一般流程



Geometry Description matrix

decsg

Decomposed Geometry matrix

Geometry M-file

initmesh

Boundary Condition matrix

Boundary M-file

Mesh data

refinemesh

Coefficient matrix

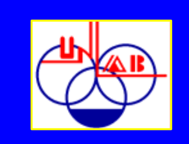
Coefficient M-file

assempde

Solution data

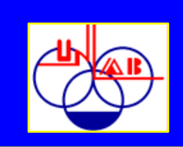
pdeplot

PDE Toolbox函数:用户界面算法



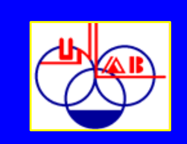
函数	目的
pdecirc	画圆
pdeellip	画椭圆
pdemdlcv	转化为1.0版本的M文件
pdepoly	画多边形
pderect	画矩形
pdetool	进入工具箱的用户界面

PDE Toolbox函数：几何算法



函数	目的
csgchk	检查几何描述矩阵的有效性
csgdel	删除最小区域之间的分界线
decsg	分解结构矩阵成最小区域
initmesh	创建初始网格
jigglemesh	调整三角形网格内的点
pdearcl	在表达式参数和弧长之间插值
poimesh	在矩形区域上创建规则网格
refinemesh	加密三角形网格
wbound	写边界条件说明文件
wgeom	写几何说明文件

PDE Toolbox函数：PDE数值计算函数



函数	目的
adaptmesh	生成自适应网格和解PDE问题
assema	组装积分区域贡献，K,M,F
assemb	组装边界条件贡献，Q,G
assempde	组装刚度矩阵和方程右边的函数
hyperbolic	解双曲型PDE问题
parabolic	解抛物型PDE问题
pdeeig	解特征值PDE问题
pdenonlin	解非线性PDE问题
poisolv	求矩形网格上泊松方程的快速解

PDE Toolbox函数:



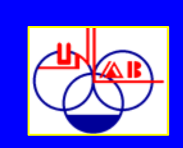
绘图函数

函数	目的
pdecont	绘制等值线图的快速命令
pdegplot	绘制PDE几何图
pdemesh	绘制PDE三角形网格图
pdeplot	一般PDE工具箱的绘图函数
pdesurf	绘制表面图的速写命令

用户定义算法

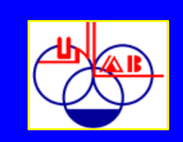
函数	目的
pdebound	绘制等值线图的快速命令
pdegplot	绘制PDE几何图

PDE Toolbox函数：工具算法



函数	目的	函数	目的
dst	离散正弦变换	pdejmps	为自适应进行误差估计
pdeadgsc	用相对误差判别准则选择最低劣的三角形	pdesde	子区域集中边缘的索引
		pdesdp	子区域集中点的索引
pdeadworst	相对最差值选择低劣三角形	pdesdt	子区域集中三角形索引
		pdesmesh	计算结构力学张量函数
pdecgrad	计算PDE通量	pdetrg	三角形几何数据
pdeent	与已知三角形相邻的三角形的索引	pdetrip	三角形几何数据
		pdetriq	测量三角形网格质量
pdeintrp	从单元节点数据到三角形单元中点的插值	sptarm	解一般稀疏矩阵特征值问题
pdeprtni	从三角形中点数据到节点数据的插值	tri2grid	由PDE三角形网格转换成矩形网格

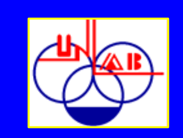
命令行求解PDE的一般步骤



1. 问题定义及参数初始化－pdetool, wbound, wgeom
   定义求解区域（几何条件）、边界条件和方程系数及参数初始化
2. 网格化（三角形网格划分、网格细化及微调）－initmesh, refinemesh, jigglemesh
   为便于观察网格，可以采用pdemesh(p,e,t)绘制由网格数据p,e,t制定的网格
3. 求解PDE－assempde, parabolic, hyperbolic, pdenonlin
4. 显示结果：pdesurf, pdeplot, pdecont

请通过Help命令查看以上命令的调用格式

命令行求解椭圆方程示例

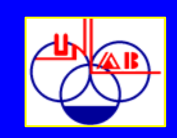


求解单位圆上的泊松方程：

$$\begin{cases} -\Delta u = 1, \Omega = \{(x,y) \mid x^2 + y^2 < 1\} \\ u|_{\partial\Omega} = 0 \end{cases}$$

```
%求解问题的几何和边界条件已分别由circleg,circleb1定义
[p,e,t]=initmesh('circleg','Hmax',1);        → 初始化网格
[p,e,t]=refinemesh('circleg',p,e,t);
[p,e,t]=refinemesh('circleg',p,e,t);         → 网格加密
[p,e,t]=refinemesh('circleg',p,e,t);
u=assempde('circleb1',p,e,t,1,0,1);
pdemesh(p,e,t)                               → 绘制网格图
figure
pdesurf(p,t,u)                               → 绘制解曲面图
```

命令行与GUI联合求解偏微分方程示例



求解单位圆上的泊松方程:

$$\begin{cases} -\Delta u = 1, \Omega = \left\{ (x, y) \middle| x^2 + y^2 < 1 \right\} \\ u|_{\partial\Omega} = 0 \end{cases}$$

1. 利用GUI定义几何模型和边界条件;
2. 在GUI的[Boundary]菜单中，选择[Export Decomposed Geometry, Boundary Cond’s...]，将几何模型和边界条件输出
3. 执行>>wbound(b,’poissonb’)，和>>wgeom(g,’poissong’)，将几何模型和边界条件存入文件poissong.m和poissonb.m
4. 采用以下语句求解方程并输出结果

```
g='poissong';
b='poissonb';
c=1;a=0;f=1;
[p,e,t]=initmesh(g);
[p,e,t]=refinemesh(g,p,e,t);
u=assempde(b,p,e,t,c,a,f);
pdesurf(p,t,u)
colormap(cool)
```

等温一级反应的圆柱状催化剂颗粒中浓度分步



$$\begin{cases} \frac{1}{R}\frac{\partial}{\partial R}\left(R\frac{\partial C}{\partial R}\right)+\left(\frac{D}{L}\right)^2\frac{\partial^2 C}{\partial Z^2}=\phi^2 C \\ R=0,-1\le Z\le 1,\frac{\partial C}{\partial R}=0 \\ R=1,-1\le Z\le 1,C=1 \\ Z=\pm 1,0\le R\le 1,C=1 \end{cases}$$

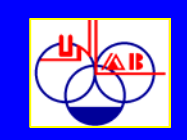
$$x\equiv R, y\equiv \frac{L}{D}Z, u\equiv C$$

$$\begin{cases} -\nabla\cdot x\nabla u = x\phi^2 u \\ x=0,-\frac{L}{D}\le Z\le \frac{L}{D},\frac{\partial u}{\partial x}=0 \\ x=0,-\frac{L}{D}\le Z\le \frac{L}{D},u=1 \\ y=\pm\frac{L}{D},0\le x\le 1,u=1 \end{cases}$$

$$\phi=4, L=1, D=0.5$$

```
function CylindCat1
g = 'CylindCat1g';
b = 'CylindCat1b';
c = 'x';a = 0; f = '-4*x.*u';
[p,e,t] = initmesh(g);
[p,e,t] = refinemesh(g,p,e,t);
p = jigglemesh(p,e,t);
[u,res] = pdenonlin(b,p,e,t,c,a,f)
pdesurf(p,t,u),colorbar
xlabel('x'),ylabel('y'),zlabel('u')
```

利用Matlab求解偏微分方程的策略



1. 对于一维动态模型，优先使用Matlab函数pdepe
2. 对于二维稳态和动态模型
   - 含有两个方程以下的标准PDE问题，优先采用pdetool
   - 含有三个方程以上的标准PDE问题，选择PDE工具箱函数
3. 当以上方法均无法解决问题，考虑使用正交配置法，有限差分等方法。对于许多化工偏微分方程问题而言，MOL法也是不错的方法

第七讲
应用数理统计和最优化方法

化工学院软件应用教科组

2006-11

化工数学模型建立的一般步骤



模型化

参数估计
模型辨识

数据处理

数据处理

最优化方法

实验

实验设计

数理统计

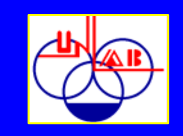
实验设计

本章知识要点



- Matlab统计工具箱函数
- Matlab最优化工具箱函数
- Matlab遗传算法与直接搜索工具箱
- 最小二乘法线性、非线性曲线拟合

Statistics Toolbox的内容分类



- 概率分布（Probility Distribution）
- 描述统计（Descriptive Statistics）
- 统计可视化（Statistical Visualization）
- 假设检验（Hypothesis Tests）
- 线性模型（Linear Models）
- 非线性回归（Nonlinear Regression Models）
- 多元分析（Multivariate Statistics）
- 统计的过程控制（Statistical Process Control）
- 实验设计（Design of Experiments）
- 隐马尔科夫模型（Hidden Markov Models）

随机变量与概率分布



- 随机变量：按一定概率可以在一个特定数集中取值的变量
- 概率分布：一个随机变量取给定值或属于一给定值集的概率所确定的函数
  - 分布函数：$F(x) = P(X \le x)$
  - 概率密度函数：$f(x) = dF(x)/dx$ $P(x = x_i) = p_i$
  - 分布函数的逆函数也称为分位数：$F(x_p) = p$

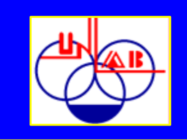
Statistics Toolbox中的概率分布种类



- Matlab提供约30种概率分布，主要的如下:
  - 二项分布（binomial distribution）
  - 正态分布（normal distribution）
  - 泊松分布（poisson distribution）
  - 卡方分布（Chi-square distribution）
  - t分布　　　（student distribution）
  - F分布　　　（F-distribution）
  - 指数分布　（exponential distribution）

  ……

概率分布的功能函数



- 对于每种概率分布，Matlab提供了如下6种功能函数
  - 概率密度函数；*pdf
    (normpdf, chi2pdf, betapdf,binopdf, tpdf, fpdf...)
  - 累积分布函数；*cdf
    (normcdf, chi2cdf, betacdf,binocdf, tcdf, fcdf...)
  - 分位数函数；*inv
  - 分布统计函数；*stat
  - 分布拟合函数；*fit
  - 随机数生成函数；*rnd

正态分布的计算



累积分布函数normcdf:

例：标准正态分布在区间(-1,1)内的取值概率

```
p=normcdf([-1 1],0,1); p(2)-p(1)
```

累积分布函数normpdf:

例：标准正态分布,x=1的概率密度

```
p=normpdf(1,0,1)
```

分位数函数norminv:

例：标准正态分布的包含95%取值的区间

```
x=norminv([0.025 0.975],0,1)
```

可通过disttool打开分布函数图形界面，观察计算结果

描述性统计



- 位置特征
  - 算术平均， mean
  - 中位数， median
  - 切尾平均， trimmean
  - 几何平均， geomean
  - 调和平均， harmmean
  - 众数， mode
- 变异特征
  - 极差， range
  - 平均绝对偏差， mad
  - 方差， var
  - 标准差， std
  - 四分位极差， iqr

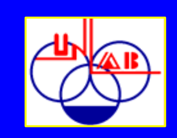
假设检验



- 假设检验是一种为了确定一个有关总体分布的假设应该予以拒绝还是接受的程序。
- Statistics Toolbox假设检验函数包括以下两类：

- 统计检验
  - dwtest
  - ranksum
  - runstest
  - signrank
  - signtest
  - ttest
  - ttest2
  - vartest
  - vartest2
  - ztest
- 分布检验
  - chi2gof
  - jbtest
  - kstest
  - kstest2
  - lillietest

线性模型 – 线性回归



- 多元线性模型：$$y = X\beta + \varepsilon$$ $$\varepsilon \sim N(0, \sigma^2 I)$$

- 多元线性回归函数：regress

```
b = regress(y,X)
[b,bint] = regress(y,X)
[b,bint,r] = regress(y,X)
[b,bint,r,rint] = regress(y,X)
[b,bint,r,rint,stats] = regress(y,X)
[b,bint,r,rint,stats] = regress(y,X,alpha)
```

其中：

b，回归系数；bint，回归系数的置信区间

r，残差向量；rint，残差向量的置信区间

stats包含四个值：相关系数、F统计量值相应于F统计量值的概率、误差方差估计

alpha，置信水平

regress的使用

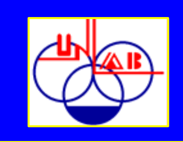


y	x1	x2	x3
4.3002	2	18	50
3.6485	7	9	40
4.4830	5	14	46
5.5468	12	3	43
5.4970	1	20	64
3.1125	3	12	40
5.1182	3	17	64
3.8759	6	5	39
4.6700	7	8	37
4.9536	0	23	55
5.0060	3	16	60
5.2701	0	18	49
5.3772	8	4	50

已知某观测量y的值与x1,x2,x3三个变量有关，如左表所示，采用线性回归拟合函数

```
y=[4.3302 3.6485 4.4830 5.5468 5.4970 3.1125...
 5.1182 3.9759 4.6700 4.9536 5.0060 5.2701 5.3772]';
x1=[2 7 5 12 1 3 3 6 7 0 3 0 8];
x2=[18 9 14 3 20 12 17 5 8 23 16 18 4];
x3=[50 40 46 43 64 40 64 39 37 55 60 49 50];
x0=ones(1,13);
X=[x0;x1;x2;x3]';
[b,bint,r,int,stats]=regress(y,X)
```

多项式回归



- 多项式回归的主要函数

  1.[a,S]=polyfit(x,y,n)

  2.[y,DeltaA]=polyval(a,x,S)

  3.[y,DeltaA]=polyconf(a,x,S,alpha)

  - 如果输入数据的误差相互独立，且有常数方差的正态分布，则y ± DeltaA包含100(1-alpha)%的置信区间

  4.ploytool(x,y,n,alpha)

  - 给出用n阶多项式拟合x,y数据，并用描绘100(1-alpha)%置信区间

除线性回归和多项式回归外，统计工具箱中线性回归的分类中还包括二次响应面模型、广义线性回归、逐步回归

线性模型－方差分析



- 方差分析(Analysis of Variance,ANOVA)是研究一些因素（自变量）对某个指标（因变量）的相关关系，研究因素对指标的影响是否显著

  - 单因素实验方差分析函数：anova1
  - 双因素实验方差分析函数：anova2
  - 多因素实验方差分析函数：anovan

多元统计分析



- 主成分分析：考虑多个数值变量间相关性的一种多元统计方法
  - –princomp，pcacon，pcares，barttest
- 聚类分析：多元统计中研究如何“物以类聚”的统计方法
  - –clusterdata，pdist，squareform，linkage
- 判别分析：研究如何根据已知归属类型的一批数据来推断未知归属类型的数据中每个个体所属类型的一种多元统计方法
  - –classify，mahal
- 多元方差分析
  - –manova，manovacluster

实验设计



- 完全析因设计（Full Factorial Desing）
  - –fullfact，ff2n
- 不完全析因设计（Fractional Factorial Design）
  - –fracfact
- 响应面设计（Response Surface Design）
  - –bbdesign，ccdesign
- D-优化设计（D-Optimal Design）
  - –manova，manovacluster

最优化问题的一般形式



目标函数 $$\min_{x\in R^n} J = f(x)$$ → 线性优化

非线性优化

二次优化

多目标优化

等式约束方程 $$G(x_i)=0, i=1,2,\cdots.m_e$$

不等式约束方程 $$G(x_i)\le 0, i=m_e+1,\cdots,m$$

参数边界 $$x_l \le x \le x_u$$

Matlab最优化函数分类



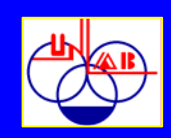
- 求极小值问题
  - 非线性最小化（Nonlinear minimization of functions）
  - 多目标非线性最小化（Nonlinear minimization of multi-objective functions）
- 最小二乘问题
  - 线性最小二乘（Linear least squares of matrix problems)
  - 函数的最小二乘（Nonlinear least squares of functions)
- 方程求解（Equation solving)
  - fzero,fsolve,\
- 优化函数控制

极小值问题函数



函数	功能
bintprog	二元整数规划
fgoalattain	多目标达到问题
fminbnd	有边界的单变量非线性最小化问题
fmincon	有约束非线性最小化问题
fminmax	最小最大问题
fminsearch	无约束非线性最小化
fminunc	无约束非线性最小化
fseminf	半无限问题
linprog	线性规划问题
quadprog	二次规划问题

函数fminbnd和单变量最优化问题



数学模型: $\min_{x} f(x)$ $x_1 < x < x_2$

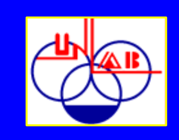
调用格式:

```
x = fminbnd(fun,x1,x2)
x = fminbnd(fun,x1,x2,options,…p1,p2)
[x,fval] = fminbnd(...)
[x,fval,exitflag] = fminbnd(...)
[x,fval,exitflag,output] = fminbnd(...)
```

输入参数:

fun	目标函数，可以由M文件定义，或由匿名函数确定
x1，x2	求解区间[x1,x2]
options	优化参数选项
p1，p2	需要传递的参数

函数fminbnd和单变量最优化问题



输出参数:

x　　　　最优解

fval　　　　目标函数的最小值

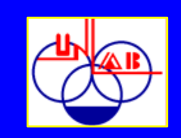
exitflag　　　　描述退出条件:

- exitflag>0　　　　表示目标函数收敛于解x处;
- exitflag = 0　　　　表示已经达到函数评价或迭代的最大次数;
- exitflag<0　　　　表示目标函数不收敛

output　　　　是一个结构体，输出优化过程的各种信息

- output.algorithm　　　　使用的算法;
- output.func　　　　函数评价次数
- output.iterations　　　　迭代次数
- output.message　　　　退出信息

函数fminbnd的使用

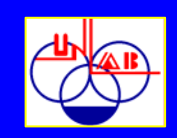


在区间[0 100]内求取单变量函数的最小值

$$\min f(x) = x^3 + 3x^2 - 9x$$

```
function Cha7demo1
x1 = 0;
x2 = 100;
[x,fval] = fminbnd(@ObjFunc,x1,x2);
fprintf(' \nResults:\n')
fprintf('   Optimum solution: %f\n',x)
fprintf('   Objective value: %f',fval)
% ----------------------------------------------------------------
function f = ObjFunc(x)
f = x^3 + 3*x^2 - 9*x;
```

函数linprog和线性规划



$$\min_x f^T x \quad \text{such that} \quad \begin{cases} A \cdot x \le b \\ Aeq \cdot x = beq \\ lb \le x \le ub \end{cases}$$

调用格式:

```
x = linprog(f,A,b)
x = linprog(f,A,b,Aeq,beq)
x = linprog(f,A,b,Aeq,beq,lb,ub,x0,options)
[x,fval] = linprog(...)
[x,fval,exitflag,output,lambda] = linprog(...)
```

其中lambda是一个结构体，含有在解x处使用的Lagrange乘子。其余与前类似

函数linprog的使用

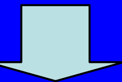


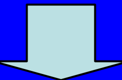
$\max \quad -5x_1 + 6x_2 + 7x_3$

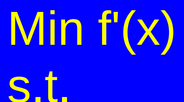
$$s.t. \quad \begin{cases} 5x_1 - 6x_2 + 10x_3 \le 20 \\ -x_1 - 5x_2 + 3x_3 \le -15 \\ x_1 + x_2 + x_3 = 5 \\ x_1, x_2, x_3 \ge 0 \end{cases}$$



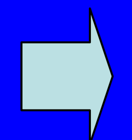
```
function Cha7demo2
f = [5 -6 -7]';
A = [5  -6  10
     -1 -5  3 ];
b = [20 -15];
Aeq = [1 1 1];
beq = [5];
lb = zeros(3,1);
[x, fval, exitflag,output,lambda] = linprog(f,A,b,Aeq,beq,lb)
```

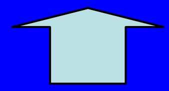






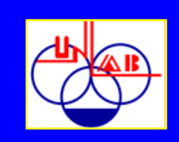
Min f'(x)
s.t. AX <= B
AeqX = Beq
LB <= X <= UB





min 5x1 - 6x2 - 7x3
s.t. 5x1 - 6x2 + 10x3 <= 20
-x1 - 5x2 + 3x3 <=-15
x1 + x2 + x3 = 5
x1,x2,x3 >=0

函数fminunc，fminsearch和无约束多变量问题最优化



数学模型： $\min_{x} f(x)$ 其中，x为一向量

fminunc和fminsearch分别采用的是拟牛顿法和Nelder-Mead单纯形法

调用格式:

x = fminunc(fun,x0,options…p1,p2)

[x,fval,exitflag,output,grad,hessian] = fminunc(...)

fminsearch与fminunc调用格式相同。

函数fmincon和多变量有约束最优化问题



数学模型：

$$\min_{x} f(x) \quad \text{subject to}$$

$$c(x) \le 0$$
$$ceq(x) = 0$$
$$A \cdot x \le b$$
$$Aeq \cdot x = beq$$
$$lb \le x \le ub$$

调用格式：

```
x = fmincon(fun,x0,A,b)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(...)
```

函数fmincon的使用



$$\min_x f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$$

s.t.

$$x_1 - x_2 \le 1$$
$$x_1 + x_2 = 0$$
$$1.5 + x_1x_2 - x_1 - x_2 \le 0$$
$$-x_1x_2 - 10 \le 0$$

```
function Cha7demo4
x0=[-1,1];a=[1,-1];b=1;a1=[1,1];b1=0'
[x,f]=fmincon(@objfun,x0,a,b,a1,b1,[],[],@nonlcon)
%------------------------------------------------------
function f = objfun(x)
f = exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
%-----------------------------------------------------
function [c1,c2]=nonlcon(x)
c1=[1.5+x(1)*x(2)-x(1)-x(2);-x(1)*x(2)-10];
c2=0;
```

最小二乘法问题



线性最小二乘（Linear least squares of matrix problems)）	
lsqlin	线性约束的最小二乘
lsqnonneg	非负约束的最小二乘
函数的最小二乘（Nonlinear least squares of functions)）	
lsqcurvefit	最小二乘法曲线拟合
lsqnonlin	非线性最小二乘问题

lsqnonlin可以求解问题的模型如下:

$$\min_x \frac{1}{2}\|F(x)\|_2^2 = \frac{1}{2}\sum_i f_i(x)^2$$

lsqcurvefit可求解问题的模型如下:

$$\min_x \frac{1}{2}\|F(x,xdata)-ydata\|_2^2 = \frac{1}{2}\sum_{i=1}^{m}(F(x,xdata_i)-ydata_i)^2$$

lsqcurvefit，lsqnonlin的使用



调用格式:

```
x = lsqnonlin(fun,x0,lb,ub,options)
[x,resnorm,residual,exitflag,output,lambda,jacobian] = lsqnonlin(...)
x = lsqcurvefit(fun,x0,xdata,ydata,lb,ub,options)
[x,resnorm,residual,exitflag,output,lambda,jacobian] = lsqcurvefit(...)
```

输入参数:

fun　　定义F(x)，注意函数返回的是fun(x)

x0　　初值

lb，ub　　设计变量的下边界和上边界，如果X(i)下无界，则设置lb = －inf，如果上无界，则ub = inf。

输出参数:

x　　拟合系数

resnorm　　x处的残差平方和范数值

residual　　x处的残差值

jacobian　　x处的Jacobian矩阵

最小二乘法非线性曲线拟合



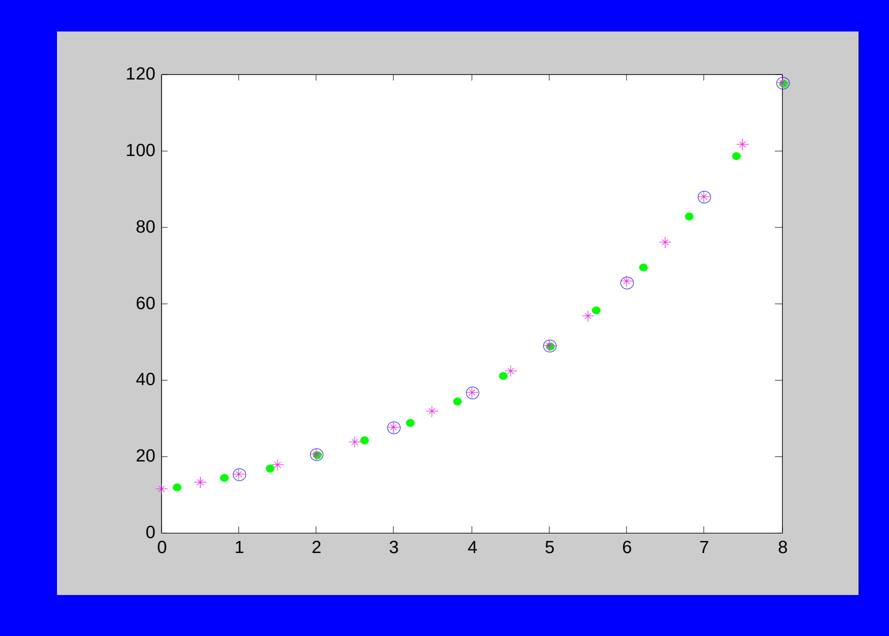
已知数据：

X1	1	2	3	4	5	6	7	8
Y1	15.3	20.5	27.4	36.6	49.1	65.6	87.8	117.6

拟合如下函数形式的曲线:

$$y = a_1 e^{a_2 x}$$

```
function  Cha7demo5
global x y
x=1:8;
y=[15.3 20.5 27.4 36.6 49.1 65.6 87.8 117.6];
a0=[1,1];
[a,resnorm,residual]=lsqcurvefit(@objfun,a0,x,y)
a2=lsqnonlin(@objfun2,a0)
x1=[0.2:0.6:8];
y1=a(1)*exp(a(2)*x1)
x2=[0:0.5:8];
y2=a2(1)*exp(a2(2)*x2);
plot(x,y,'bo',x1,y1,'g.',x2,y2,'m*')
%---------------------------------------------
function f=objfun(a,x)
f=a(1)*exp(a(2)*x);
%---------------------------------------------
function z=objfun2(a,x)
global x y
for i=1:8
    z(i)=a(1)*exp(a(2)*x(i))-y(i);
end
```



置信区间



nlparci: 非线性回归参数的置信区间

```
ci = nlparci(beta,'covar',sigma)
ci = nlparci(beta,resid,'jacobian',J)
ci = nlparci(...,‘alpha’, alpha）
```

nlpredci: 非线性回归预测值的置信区间

```
[ypred, delta] = nlpredci(modelfun,x,beta,resid,'covar',sigma)
[ypred, delta] = nlpredci(modelfun,x,beta,resid,'jacobian',J)
[...] = nlpredci(...,'param1',val1,'param2',val2,...)
```

nlintool: 以两条红线绘制预测值的置信区间

```
nlintool(x,y,fun,beta0)
```

优化函数的控制



optimget: 得到最优化函数选项参数值

optimset: 创建或编辑最优化函数选项参数值

调用格式:

opts=optimset('para1','value1','para2','value2',...)

将指定参数para赋为的value

optimset

返回所有可以供optimset设置的参数值

opts=optimset

创建一个结构，所有的值都被赋予[]

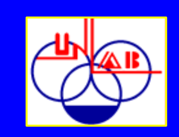
opts=optimset(oldopts,'para1','value1',...)

从oldopts复制参数值，并修改相应的参数值

opts=optimset(oldopts,newopts)

oldopts和newopts组合，以newopts的非零参数值覆盖oldopts中的对应值

遗传算法与直接搜索工具箱简介

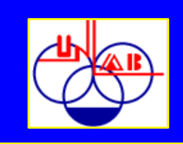


Matlab7.0 R14以后的版本中包含了一个专门设计的遗传算法与直接搜索工具箱（Genetic Algorithm and Direct Search Toolbox，GADST）

拓展了原有最优化工具箱的功能，可以求解普通方法不易求解的问题，如查找问题

与最优化工具箱紧密结合，可以充分发挥工具箱功能，以提高求解的质量。对于某些问题可以获得全局最优解

遗传算法的基本概念



- 遗传算法是一种模拟自然选择、生物进化过程来求解问题的方法
- 算法要点

  1.创建一个随机种群(Population)

  2.生成一个新的种群

  a) 通过计算适应度(Fitness)，给当前种群的每一个个体打分

  b) 根据适应度选择父辈

  c) 由父辈产生子辈，可以通过复制(Reproduction)、交叉(Crossover)、变异(Mutation)和迁移(Migration)进行

  d) 用子辈替换当前种群，形成下一代(Generation)

  3.若满足停止准则，则算法停止

遗传算法工具GUI



- 在命令窗口键入gatool，可打开遗传算法GUI求解问题

Genetic Algorithm Tool

File Help

Fitness function:
Number of variables:

Constraints:
Linear inequalities: A = b =
Linear equalities: Aeq = beq =
Bounds: Lower = Upper =
Nonlinear constraint function:

Plots
Plot interval: 1
Best fitness | Best individual | Dista
Expectation | Genealogy | Range
Score diversity | Scores | Selec
Stopping | Max constraint
Custom function:

Run solver
Use random states from previous run
Start | Pause | Stop
Current generation:
Status and results: Clear Statu
Final point:

Options:
Population
Population type: Double Vector
Population size: 20
Creation function: Uniform
Initial population: []
Initial scores: []
Initial range: [0 ; 1]
Fitness scaling
Selection
Reproduction
Mutation
Crossover
Migration
Algorithm settings
Hybrid function
Stopping criteria
Output function
Display to command window
Vectorize

Quick reference:

Genetic Algorithm Solver

This tool corresponds to the ga function.

Fitness function (required) is the objective function you want to minimize. You can specify the function as a function handle of the form @objfun, where objfun.m is an M-file that returns a scalar.

Number of variables (required) is the number of independent variables for the fitness function.

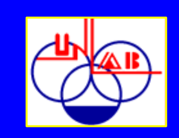
- Constraints
- Plot Functions
- Run solver

Options

Specify options for the Genetic Algorithm solver.

- **Population**
- **Fitness scaling**
- **Selection**
- **Reproduction**
- **Mutation**

遗传算法工具GUI



$$\min_{x} f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$$

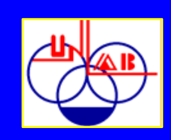
s.t.

$$x_1 - x_2 \le 1$$
$$x_1 + x_2 = 0$$
$$1.5 + x_1x_2 - x_1 - x_2 \le 0$$
$$-x_1x_2 - 10 \le 0$$

```
function f = ga_objfun1(x)
 f  = exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
```

```
function [c1,c2]=ga_nonlcon1(x)
  c 1=[1.5+x(1)*x(2)-x(1)-x(2);-x(1)*x(2)-10];
  c 2=0;
```

函数ga的用法



调用格式：

x = ga(fitnessfun, nvars, options)
x = ga(fitnessfcn,nvars,A,b,Aeq,beq,LB,UB,nonlcon,options)
[x, fval, reason, output, population, scores] = ga(...)

数学模型：

$$\min_{x} f(x) \quad \text{subject to}$$

$$c(x) \le 0$$
$$ceq(x) = 0$$
$$A \cdot x \le b$$
$$Aeq \cdot x = beq$$
$$lb \le x \le ub$$

输入参数：

fitnessfun	目标函数
nvars	解向量的长度
A,b	线性不等式约束
Aeq，beq	线性等式约束
LB，UB	自变量的下、上限
nonlcon	非线性约束
options	gaoptimset设置的算法参数

输出参数：

reason	表明算法终止的字符
output	表明每代计算结果和算法效果的结构体，

randstate：随机数发生器的状态
randnstate：均匀分别随机数发生器状态
generations：计算代数
funccount：适应度函数的计算次数
message：算法的终止原因

population	每列是最终的种群
scores	最终种群的评价

函数ga()的用法



$$\min_x f(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$$

s.t.

$$\begin{aligned} & x_1 - x_2 \le 1 \\ & x_1 + x_2 = 0 \\ & 1.5 + x_1x_2 - x_1 - x_2 \le 0 \\ & -x_1x_2 - 10 \le 0 \end{aligned}$$

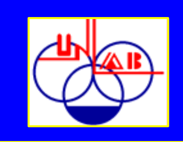
```
function Cha7demo4
x0=[-1,1];a=[1,-1];b=1;a1=[1,1];b1=0;
[x,f]=fmincon(@objfun,x0,a,b,a1,b1,[],[],@nonlcon)
opt=gaoptimset('PlotFcns',{  @gaplotbestf @gaplotbestindiv })
[x2,f2,reason,output,pop,score]=ga(@objfun,2,a,b,a1,b1,[],[],@nonlcon,opt)
[x3,f3]=fmincon(@objfun,x2,a,b,a1,b1,[],[],@nonlcon)
%-------------------------------------------------------
function f = objfun(x)
f = exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
%------------------------------------------------------
function [c1,c2]=nonlcon(x)
c1=[1.5+x(1)*x(2)-x(1)-x(2);-x(1)*x(2)-10];
c2=0;
```

遗传算法参数设置



1. 图形参数
2. 种群参数－种群多样性增加，种群个体增加可能提高结果质量
3. 适应度计算参数－适应度测量值变化大，则高测量值个体复制快，会影响搜索空间的大小
4. 选择参数
5. 再生参数－优良个体多则可以使最适应个体控制种群，但可能减少搜索有效性
6. 变异参数－变异和交叉是遗传算法的两个主要过程，但参数的设置需通过一定实验寻找最佳值
7. 交叉参数
8. 迁移参数
9. 算法设置
10. 混合函数参数－可将遗传算法与其它算法的结合，提高结果
11. 停止条件参数
12. 输出函数参数
13. 显示到命令窗口参数
14. 向量参数

直接搜索工具箱



- 与遗传算法相同，直接搜索也是一种全局优化算法
- 在命令窗口键入psearchtool，可打开模式搜索GUI求解问题，其使用方法与遗传算法工具箱类似
- 直接搜索算法的核心函数是patternsearch

模型化前数据的准备



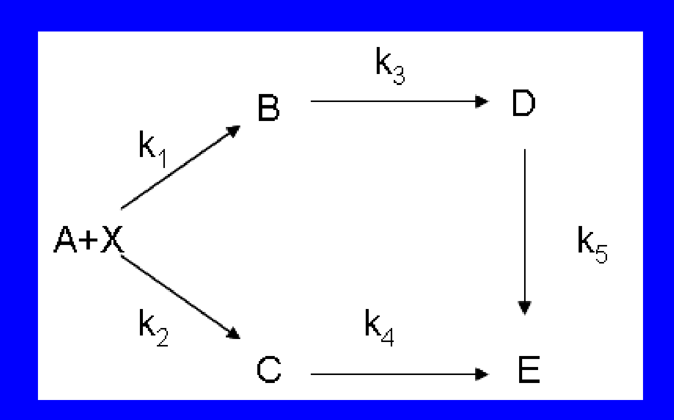
- 采用图表直观观察数据是否有异常，对于异常数据点应舍弃，必要时应重复实验
- 数据是否一致
- 数据是否不足
- 应重视实验设计

间歇反应器中串联－平行复杂反应系统



在间歇反应器中进行液相反应制备产物B，反应网络如图5-1所示。反应可在180~260℃的温度范围内进行，反应物X大量过剩，而C, D和E为副产物。各反应均为一级动力学关系：r＝－kC，式中 $k = k_0 e^{\frac{-Ea}{RT}}$

已知参数：k01＝5.78052×1010，k02＝3.92317×1012，k03＝1.64254×104，k04＝6.264×108，Ea1=124670，Ea2=150386，Ea3=77954，Ea4=111528。初始浓度：C_A=1kmol/m3，其余物质浓度为0。用最优化方法计算使产物B收率最大的最优反应温度。



间歇反应器中串联－平行复杂反应系统



数学模型

$$\frac{dC_A}{dt} = -(k_1 + k_2)C_A$$

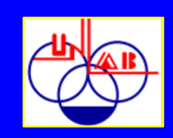
$$\frac{dC_B}{dt} = k_1 C_A - k_3 C_B$$

$$\frac{dC_C}{dt} = k_2 C_A - k_4 C_C$$

$$\frac{dC_D}{dt} = k_3 C_B - k_5 C_D$$

$$\frac{dC_E}{dt} = k_4 C_C + k_5 C_D$$

间歇反应器中串联－平行复杂反应系统



```
function Cha7demo6
R = 8.31434;                          % Gas constant, kJ/kmol K
k0 = [5.78052E+10  3.92317E+12  1.64254E+4  6.264E+8];
Ea = [124670  150386  77954  111528];   % Activation energy, kJ/kmol
C0 = [1  0  0  0  0];   % Initial Concentration,C0(i), kmol/m^3
tspan = [0 1e4];
T0 = 200+273;          % Reactor temperature, Kelvin
T = fminsearch(@ObjFunc,T0,[],tspan,C0,k0,Ea,R);
fprintf('\t使产物B收率最大的最优反应温度为%3.1f℃\n',T - 273)
% ------------------------------------------------------------
function f = ObjFunc(T,tspan,C0,k0,Ea,R)
[t,C] = ode45(@MassEquations, tspan, C0,[],k0,Ea,R,T);
CBmax = max(C(:,2));
f = - CBmax;
% ------------------------------------------------------------
function dCdt = MassEquations(t,C,k0,Ea,R,T)
k = k0.*exp(-Ea/(R*T));
k(5) = 2.16667E-04;
rA = -(k(1)+k(2))*C(1);
rB = k(1)*C(1)-k(3)*C(2);
rC = k(2)*C(1)-k(4)*C(3);
rD = k(3)*C(2)-k(5)*C(4);
rE = k(4)*C(3)+k(5)*C(4);
dCdt = [rA; rB; rC; rD; rE];
```

乙烯加氢反应动力学参数估计



乙烯（E）加氢（H）制乙烷（EA）反应如下：$H_2+C_2H_4\rightarrow C_2H_6$

微分反应器中测得的实验数据如下：

实验号	反应速率	p_E/atm	p_{EA}/atm	p_H/atm
1	1.0400	1.0000	1.0000	1.0000
2	3.1300	1.0000	1.0000	3.0000
3	5.2100	1.0000	1.0000	5.0000
4	3.8200	3.0000	1.0000	3.0000
5	4.1900	5.0000	1.0000	3.0000
6	2.3910	0.5000	1.0000	3.0000
7	3.8670	0.5000	0.5000	5.0000
8	2.1990	0.5000	3.0000	3.0000
9	0.7500	0.5000	5.0000	1.0000

反应速率方程为：

$$-r_A=\frac{kp_E\,p_H}{1+K_E\,p_E}$$

根据表中数据确定动力学参数k和K_E

```
function  Cha7demo7
rA = [1.04 3.13 5.21 3.82 4.19 2.391 3.867 2.199 0.75];
Pe = [1 1 1 3 5 0.5 0.5 0.5 0.5];
Ph = [1 3 5 3 3 3 5 3 1];
% 线性拟合
RA = Pe.*Ph./rA; y = RA'; X = [ones(size(y))  Pe'];
b = X\y; k = 1/b(1);
KE = b(2)*k;
% 非线性拟合
beta0 = [k KE]
[beta,resnorm,residual,exitflag,output,lambda,jacobian] = ...
    lsqnonlin(@ObjFunc,beta0,[],[],[],rA,Pe,Ph)
ci = nlparci(beta,residual,jacobian)
% 残差关于拟合值的残差图
rAc = RateA(beta,Pe,Ph)
plot(rAc,residual,'*')
xlabel('反应速率（-rA）拟合值, mol/(kg cat. s)')
ylabel('残差R, mol/(kg cat. s)')
refline(0,0)
 % 参数辨识结果
fprintf('Estimated Parameters:\n')
fprintf('\tk = %.3f ¡À %.3f\n',beta(1),ci(1,2)-beta(1))
fprintf('\tKE = %.3f ¡À %.3f\n',beta(2),ci(2,2)-beta(2))
fprintf('\tThe sum of the squares is: %.3f',resnorm)
% ------------------------------------------------------------
function f = ObjFunc(beta,rA,Pe,Ph)
f = rA - RateA(beta,Pe,Ph);
% ------------------------------------------------------------
function rA = RateA(beta,Pe,Ph)
rA = beta(1)*Pe.*Ph./(1+beta(2)*Pe);
```

Thank you!

祝大家在今后的学习生活中一切顺利!